

PROGETTO BASI DI DATI A.A 2021/2022

Reselling Online

Prof. Maratea Antonio



Realizzato da:

Vincenzo Salzano Mat: 0124002051

Simone D'Angelo Mat: 0124002601

INDICE

CAPITOLO 1.....	1
Informazioni Generali.....	1
Realtà d'uso.....	2
Descrizione del Progetto.....	4
CAPITOLO 2.....	6
Analisi entità.....	6
Cliente.....	6
Carta.....	7
Acquisto.....	7
Promo.....	8
Abbonamento.....	8
Release Day.....	9
Catalogo.....	10
Prodotto.....	11
Scarpe.....	11
Capi Hype.....	12
Taglia Scarpe.....	13
Fornitore.....	14
Fornitura.....	14
Magazzino.....	15
Glossario.....	16
CAPITOLO 3.....	18
Analisi Relazioni.....	18
Molteplicità e Totalità.....	18
Associazioni N a M.....	23
CAPITOLO 4.....	24
Prestazioni attese,numero potenziale utenti.....	25
Dimensione Tabelle.....	25
Categorie Utenti.....	26
Tavola dei volumi.....	27
Privilegi d'accesso.....	28
CAPITOLO 5.....	29
Schemi scheletri di settore.....	29

Rappresentazione grafica diagramma EE/R.....	33
CAPITOLO 6.....	34
Diagramma Relazionale.....	34
CAPITOLO 7.....	35
Elenco delle operazioni.....	35
Operazioni di base.....	35
Operazioni degli utenti.....	38
Diagramma Uml Use Case.....	39
CAPITOLO 8.....	40
Vincoli d'integrità statici.....	40
Vincoli d'integrità dinamici.....	42
Vincoli d'integrità referenziale.....	43
CAPITOLO 9.....	48
Normalizzazione.....	48
CAPITOLO 10.....	49
Creazione Tabelle.....	49
Popolamento.....	55
Creazione degli utenti.....	82
Triggers.....	84
Scheduler	91
Stored Procedures.....	92
Query.....	100
Viste.....	102
CAPITOLO 11.....	106
Conclusione.....	106
Spunti e riferimenti.....	106
Ringraziamenti.....	107

CAPITOLO 1

Informazioni Generali

Si vuole realizzare una base di dati per la gestione di un sito di 'Reselling Online'.

Un sito di reselling online si occupa della gestione dei capi 'Hypebeast' (capi d'abbigliamento in edizione limitata) acquistati ad un prezzo inferiore durante la release (rilascio ufficiale sul mercato del prodotto), o da un altro rivenditore, per poi essere rivendute successivamente ad un prezzo remunerativo.

La nostra base di dati gestisce solo capi e scarpe e, su quest'ultimi, gestisce l'esclusività che questi prodotti ne comportano.

Ogni cliente per accedere esegue il login venendo contraddistinto da un Username, per poi compiere diverse azioni oltre all'acquisto di un prodotto; quest'ultimo può effettuare un abbonamento, che garantirà successivamente dei privilegi, come:

- L'accesso alle varie Release Day
- Accesso a diversi cataloghi

Nel caso in cui il cliente scegliesse di non effettuare abbonamenti, gli verrà garantito l'abbonamento 'free', per l'accesso ad alcuni cataloghi, e di conseguenza ad alcuni prodotti.

Il cliente può inserire anche più carte sul sito per poi effettuare un acquisto e, su quest'ultime, potrà controllare gli acquisti fatti.

Sugli acquisti può essere applicata una promo, che verrà presentata sul sito periodicamente, o assegnata di diritto ad un cliente.

La base di dati si occupa di gestire, oltre alla parte che contraddistingue un sito di reselling, anche la parte logistica e su dove reperire un prodotto.

Vengono quindi memorizzati tutti i fornitori delle aziende da cui possiamo comprare i prodotti, spediti in blocchi di forniture da cui andremo a prendere il singolo prodotto.

Ogni fornitura viene spedita in uno o più magazzini, nei quali verranno smistate e gestite.

Altro elemento identificativo di un sito di reselling è il prezzo. Quest'ultimo varia in base alla taglia, di solito ad avere un prezzo maggiore sono le taglie più richieste o quelle più alte, e nella nostra base di dati ci occupiamo di gestirlo per le scarpe. Il prezzo di un capo di abbigliamento Hypebeast inoltre varia di una percentuale (che fa perdere o guadagnare valore alla scarpa) in base al tempo, come se fosse una vera azione quotata in Borsa.

Realtà d'uso

Acquistare e rivendere scarpe, o capi di abbigliamento, in edizione limitata con gli stessi criteri con cui si acquisterebbero e rivenderebbero azioni in Borsa: è l'essenza del reselling, un fenomeno che è emerso negli ultimi anni e si è diffuso principalmente tra le giovani generazioni, generando business e contribuendo alla nascita di alcune piattaforme mondiali.

Tra le piattaforme mondiali quella che spicca maggiormente è Stockx, la quale ha ispirato una buona parte dei siti di reselling attualmente attivi.

Nella sua forma più semplice il reselling consiste nel possedere un prodotto molto ricercato, di solito in edizione limitata, e poterlo rivendere a prezzo maggiorato a coloro che non riescono ad ottenerlo in altro modo.

Il reselling interessa in particolare il mondo delle sneakers, ma col tempo si è esteso anche ai capi, e continua ad espandersi fino a raggiungere accessori e oggetti.

Il fenomeno è nato negli Stati Uniti e si è successivamente esteso all'Europa. Nel nostro continente i mercati più importanti sono Gran Bretagna, Francia e anche Italia, dove il reselling sta registrando un crescente successo, accompagnato da alcune critiche e polemiche. In Italia cinque anni fa i collezionisti rappresentavano circa il 5% dei consumatori, mentre i cosiddetti reseller erano circa l'1%. Oggi tutti i consumatori sono potenziali reseller, grazie ai social media e a piattaforme create ad hoc.

Il reselling è un fenomeno in ascesa anche in Italia. A novembre 2020 il fenomeno "reselling" è stato riportato in auge in Italia dal "caso Lidl". L'operatore della grande distribuzione ha messo in vendita sneakers della collezione streetwear low cost a soli 13 euro nei quasi 700 punti vendita italiani. Dopo lunghe file per accaparrarsene un paio, in pochi giorni l'articolo è andato esaurito. Ad amplificare il fenomeno una semplice regola del marketing: la Lidl Fan Collection era un'edizione ultra-limitata che ha dovuto fare fronte ad una domanda altissima, innescando un fattore psicologico di corsa all'acquisto di un articolo da status symbol. Da qui la spinta al reselling online.

Descrizione del progetto

Il progetto descrive quattro macro-aree: Acquisti, Servizi, Prodotti e Logistica.

Acquisti: incentrata sulle carte, che il cliente può inserire e selezionare; gli acquisti, che il cliente può effettuare usando una carta; le promo, che possono essere sfruttate su un singolo acquisto. Di quest'area servono:

- Per la carta (Numero carta, tipo di carta, scadenza e codice di sicurezza)
- Per l'acquisto (Codice Acquisto, data acquisto, importo dell'acquisto e le spese di spedizione)
- Per la promo (Codice Promo, data inizio, data fine, lo sconto e il nome della promo)

Servizi: incentrata principalmente sull'abbonamento, il quale può dare accesso a diversi cataloghi o alle Release Day. Di quest'area servono:

- Per l'abbonamento (Nome abbonamento, durata e prezzo)
- Per il release day (Nome Release, data release e tipo di release)
- Per il catalogo (Nome catalogo, fascia d'età e genere)

Prodotti: essa ovviamente, è al centro di tutto lo schema, ed è collegata a tutte le altre macro-aree. Incentrata principalmente sulle tipologie dei prodotti disponibili e sulle proprie caratteristiche. Di quest'area servono:

- Per il prodotto (Id prodotto, data di pubblicazione e brand)

Il prodotto a sua volta si divide in:

- Capi hype (Nome del capo, prezzo, tessuto, taglia e tipologia)
- Scarpe (Nome scarpe, prezzo, materiale e modello)
- Taglia scarpe (Numero taglia, prezzo)

Logistica: incentrata principalmente sulla fornitura, la quale può essere spedita da un fornitore e verrà poi smistata e gestita nei magazzini. Di quest'area servono:

- Per la fornitura (Id fornitura, data fornitura, quantità della fornitura, taglia fornita e prezzo della fornitura)
- Per il magazzino (Numero identificativo del Magazzino, capienza e luogo)
- Per il fornitore (Azienda del fornitore, nome, cognome e recapito telefonico)

Nel progetto si è preferito scegliere di tradurre le generalizzazioni attraverso il partizionamento verticale, in quanto sono stati aggiunti ulteriori attributi alle sottoclassi.

Nel progetto sono stati utilizzati i seguenti tipi di dato:

- int e number (per gli interi)
- varchar2 (per le stringhe di caratteri)
- char (per i singoli caratteri)
- date (per le date)

CAPITOLO 2

Analisi entità

Di seguito l'analisi dettagliata di ogni entità e delle loro proprietà.

Cliente

L'entità cliente rappresenta qualsiasi persona loggata al sito e che, di conseguenza, può effettuare determinate azioni.

Essa può inserire una, o più carte, con le quali può fare degli acquisti; può effettuare uno o più abbonamenti per poi godersi le esclusive concesse da quest'ultimo.

L'entità cliente è dotata dei seguenti attributi:

- Username: nome utente che il cliente seleziona, e che lo contraddistingue. Esso rappresenta la chiave primaria di Cliente;
- Nome: nome reale del cliente;
- Cognome: cognome reale del cliente;
- Età: età del cliente, che per accedere al sito deve essere maggiorenne;
- E-mail: mail del cliente;
- Numero di telefono: numero di telefono appartenente al cliente;
- Indirizzo: attributo strutturato, composto da via, cap e città.

Carta

La carta di credito viene inserita dal cliente per poi effettuare l'acquisto. Un cliente può inserire a sua volta più carte e su di esse visionare gli acquisti eseguiti.

L'entità carta è dotata dei seguenti attributi:

- Numero Carta: rappresenta il numero identificativo posto sul retro della carta di credito. Esso rappresenta la chiave primaria di Carta;
- Tipo carta: rappresenta la tipologia di carta inserita (es: visa, paypal ecc);
- Codice di sicurezza: rappresenta il codice di sicurezza composto da tre numeri, posto sul retro della carta;
- Scadenza Carta: data di scadenza della carta inserita.

Acquisto

L'acquisto viene effettuato da un cliente con una carta. Sull'acquisto può essere applicata una promo e può contenere uno o più prodotti.

L'entità acquisto è dotata dei seguenti attributi:

- Codice Acquisto: codice identificativo dell'acquisto. Esso rappresenta la chiave primaria di Acquisto;
- Data acquisto: indica il giorno in cui è stato effettuato l'acquisto;
- Importo acquisto: specifica l'importo effettivo totale dell'acquisto (non considerando spese di spedizione ed eventuali applicazioni di un codice promo);

-Spese di spedizione: importo che indica le spese di spedizione da pagare.

Promo

La promo in un sito di reselling è molto simile ad un promo di qualsiasi sito di vendita classico, venendo presentata sul sito ad una cadenza periodica oppure venendo consegnata singolarmente ad un cliente, che ha la facoltà di scegliere se usarla o meno.

L'entità promo è dotata dei seguenti attributi:

- Codice Promo: codice identificativo della promo. Esso rappresenta la chiave primaria di Promo;
- Data inizio promo: specifica la data in cui la promo inizia;
- Data fine promo: specifica la data in cui la promo termina;
- Nome promo: indica il nome della promo;
- Sconto: indica la cifra da scontare sull'acquisto.

Abbonamento

L'abbonamento dà accesso all'utente a vari privilegi, come i differenti cataloghi, ognuno dei quali è associato ad un determinato tipo di abbonamento, e le varie release day nel caso in cui si effettui un abbonamento differente dal free.

L'entità abbonamento è dotata dei seguenti attributi:

- Nome abbonamento: che indica il nome associato ad un tipo di abbonamento. Esso rappresenta la chiave primaria di Abbonamento;
- Durata: indica la durata in mesi del tipo di abbonamento;
- Costo: indica il prezzo associato ad un determinato abbonamento.

Release Day

Il Release day è un privilegio concesso in seguito all'acquisto di un abbonamento. Il release day consiste in un'emulazione del fenomeno di 'Dropshipping' (modello di vendita grazie al quale il venditore vende un prodotto ad un utente, senza possederlo materialmente nel proprio magazzino), e dà in dotazione all'utente dei 'pass' per poter accedere al rilascio del prodotto in edizione limitata, direttamente dal sito. Di conseguenza il sito si occupa di gestirne solo l'accesso da parte dei propri clienti, garantendogli l'opportunità di reperire un capo d'abbigliamento di lusso, siccome potrebbero esserci difficoltà ad introdurre quei prodotti sul sito per via della propria esclusività.

L'entità Release Day è dotata dei seguenti attributi:

- Nome Release: nome identificativo della release, che il cliente può visionare e in seguito decidere se accedervi o meno. Esso rappresenta la chiave primaria di Release Day;

- Data Release: data che indica il giorno in cui si terrà quest'evento. Essendo un evento in cui gestiamo solo gli accessi, non è necessario inserire una data d'inizio e una di fine, essendo la data fine derivante dalla fine delle scorte durante l'evento e ciò viene gestito quindi dal brand che organizza l'evento, e non da noi.
- Tipo Release: il tipo indica la tipologia di release che si terrà, specificando il nome del prodotto, o dei prodotti, che verranno inseriti all'interno della release.

Catalogo

I cataloghi sono accessibili sempre tramite abbonamento. Tuttavia, a differenza della Release, anche l'abbonamento free garantisce l'accesso a dei cataloghi.

Ogni abbonamento garantisce l'accesso ad un massimo di sei cataloghi, che conterranno un numero predefinito di prodotti simili tra loro, che variano esclusivamente per genere e fascia d'età.

Ogni catalogo al proprio interno conterrà prodotti che altri cataloghi non contengono, ciò è stato fatto per rendere un ambiente elitario, come un sito di reselling, ancora più esclusivo e per garantire una peculiarità specifica alla nostra base di dati.

L'entità Catalogo è dotata dei seguenti attributi:

- Nome Catalogo: nome identificativo del catalogo associato ad un determinato abbonamento. Esso rappresenta la chiave primaria di Catalogo;
- Fascia d'età: indica la fascia d'età per cui il catalogo si predispone ad essere sfogliato;
- Genere: indica il genere per cui il catalogo si predispone ad essere sfogliato.

Prodotto

L'entità prodotto descrive i prodotti coinvolti nella nostra base di dati, ovvero i Capi hype e le scarpe. Quest'entità infatti prevede una specializzazione che coinvolge i prodotti citati in precedenza.

L'entità Prodotto è dotata dei seguenti attributi:

- Id prodotto: identificativo di un determinato prodotto. Esso rappresenta la chiave primaria di Prodotto;
- Brand: indica il brand di cui fa parte tale prodotto;
- Data Pubblicazione: quest'ultima indica la prima data di pubblicazione, data in cui il prodotto ha fatto la sua prima comparsa, sul nostro sito.

Scarpe

Avendo tradotto le generalizzazioni con il partizionamento verticale, la chiave di prodotto (Id Prodotto) in scarpe verrà contrassegnata sia come chiave primaria che come chiave esterna.

Le scarpe, oltre ad essere il fulcro ed il motivo per cui il reselling è diventato di tendenza, è anche il fulcro della nostra base di dati.

L'entità Scarpe è dotata dei seguenti attributi, oltre ad Id prodotto:

- Nome Scarpe: nome specifico delle scarpe. Peculiarità delle scarpe vendute al reselling è che: nel nome di una determinata scarpa, sono inserite anche la colorazione e la

tipologia di scarpa (ad esempio: se una scarpa è una scarpa alta, nel nome verrà contrassegnata come 'High').

- Materiale: indica il materiale che costituisce la scarpa;
- Prezzo: indica il prezzo di base che la scarpa ha sul sito. Come scritto in precedenza il prezzo iniziale potrà variare di una percentuale con il tempo e gli si potrà applicare su quest'ultimo, successivamente, anche il prezzo extra della taglia.
- Modello: indica il modello specifico delle scarpe (ad esempio: sneakers ecc).

Capi Hype

Avendo tradotto le generalizzazioni con il partizionamento verticale, la chiave di prodotto (Id Prodotto) in capi hype verrà contrassegnata sia come chiave primaria che come chiave esterna.

L'entità Capi Hype è dotata dei seguenti attributi, oltre ad Id prodotto:

- Nome Capo: nome specifico del capo. Come per le scarpe, anche per molti capi la colorazione è inserita all'interno del nome;
- Tessuto: indica il tessuto da cui è costituito il prodotto;
- Prezzo: indica il prezzo di base che il capo ha sul sito. Come scritto in precedenza il prezzo iniziale potrà variare di una percentuale con il tempo;
- Taglia: indica la taglia del capo. A differenza della taglia delle scarpe, non si applicano variazioni di prezzo sul nostro sito per le varie taglie;
- Tipologia: indica la tipologia specifica del capo (ad esempio: felpa ecc).

Taglia scarpe

La taglia delle scarpe è un'entità debole di scarpe e costituisce una particolarità dei siti di reselling in cui ogni taglia aumenta il prezzo, con un determinato importo, ad una scarpa.

L'entità debole Taglia scarpe è dotata dei seguenti attributi:


- Numero Taglia: indica il numero di taglia della scarpa. Esso rappresenta la chiave debole di Taglia Scarpe;
- Prezzo: indica il prezzo aggiuntivo associato alla taglia.

Segue esempio di un sito che segue la logica appena definita.

Annulla

Jordan 4 Retro

Military Black



Seleziona Size

<div>US M 5.5</div> <div>473 €</div>	<div>US M 6</div> <div>474 €</div>	<div>US M 6.5</div> <div>483 €</div>	<div>US M 7</div> <div>365 €</div>
<div>US M 7.5</div> <div>376 €</div>	<div>US M 8</div> <div>339 €</div>	<div>US M 8.5</div> <div>366 €</div>	<div>US M 9</div> <div>362 €</div>
<div>US M 9.5</div> <div>371 €</div>	<div>US M 10</div> <div>375 €</div>	<div>US M 10.5</div> <div>380 €</div>	<div>US M 11</div> <div>387 €</div>
<div>US M 11.5</div> <div>378 €</div>	<div>US M 12</div> <div>336 €</div>	<div>US M 12.5</div> <div>324 €</div>	<div>US M 13</div> <div>331 €</div>
<div>US M 14</div> <div>310 €</div>	<div>US M 15</div> <div>280 €</div>	<div>US M 16</div> <div>294 €</div>	<div>US M 17</div> <div>339 €</div>
<div>US M 18</div> <div>380 €</div>			

Fornitore

Un prodotto in un sito di reselling può essere preso il giorno della sua release o tramite un fornitore. Nella nostra base di dati ci occupiamo esclusivamente di descrivere i fornitori da cui compriamo un prodotto.

L'entità Fornitore è dotata dei seguenti attributi:

- Azienda Fornitore: indica il nome dell'azienda di cui fanno parte i prodotti distribuiti dal fornitore. Essa rappresenta la chiave primaria di Fornitore;
- Nome Fornitore: nome del fornitore;
- Cognome Fornitore: cognome del fornitore;
- Recapito telefonico: numero di telefono del fornitore.

Fornitura

La fornitura, come scritto in precedenza, viene spedita da un fornitore e contiene al proprio interno i prodotti che verranno mostrati sul sito.

L'entità Fornitura è dotata dei seguenti attributi:

- Id Fornitura: indica il codice identificativo della fornitura. Esso rappresenta la chiave primaria di Fornitura;
- Prezzo Fornitura: indica il prezzo associato ad una fornitura;
- Taglia Fornita: indica la taglia dei prodotti contenuti all'interno della fornitura. Ogni fornitura risulta contenere sempre una sola taglia;

- Quantità: indica la quantità dei prodotti contenuti all'interno della fornitura;
- Data fornitura: la data della fornitura indica la data in cui è stata acquistata la fornitura.

Magazzino

Le forniture, una volta acquistate, vengono distribuite e gestite all'interno dei magazzini.

L'entità Magazzino è dotata dei seguenti attributi:

- Numero Magazzino: indica il codice identificativo del magazzino. Esso rappresenta la chiave primaria di Magazzino;
- Luogo: indica il luogo in cui è situato il magazzino;
- Capienza: indica la capienza massima del magazzino.

Glossario

Tabella	Descrizione	Collegamenti
Cliente	Individuo che ha un legame logico con la base di dati	Abbonamento, Carta
Abbonamento	Servizio acquistabile, o fornito, al cliente	Cliente, Release Day, Catalogo
Release Day	Servizio aggiuntivo utilizzabile tramite gli abbonamenti acquistati	Abbonamento
Catalogo	Comprende i prodotti visualizzabili, e acquistabili, sul sito	Abbonamento, Prodotto
Carta	Oggetto inseribile per eseguire acquisti	Cliente, Acquisto
Acquisto	Comprende le spese ultimate e i prodotti	Carta, Promo, Prodotto
Promo	Promozione applicabile ad un acquisto	Acquisto
Prodotto	Oggetto intorno alla quale ruota l'intera base di dati	Acquisto, Fornitura, Catalogo, Scarpe, Capi Hype
Scarpe	Un determinato tipo di prodotto	Prodotto, Taglia Scarpe
Capi Hype	Un determinato tipo di prodotto	Prodotto

Taglia Scarpe	Variazione di prezzo associato al numero delle scarpe	Scarpe
Fornitura	Stock di prodotti	Prodotto, Fornitore, Magazzino
Fornitore	Colui che spedisce le forniture	Fornitura
Magazzino	Dove vengono spedite e gestite le forniture	Fornitura

CAPITOLO 3

Analisi Relazioni

Di seguito l'analisi di molteplicità e totalità.

Molteplicità e Totalità

1. *[Release Day, Abbonamento]*

Molteplicità: N a M

Un abbonamento dà accesso ad M Release Day.

Un Release Day può essere acceduto da N abbonamenti

Totalità: Presente dal lato di Release Day

Tutti i Release Day sono acceduti tramite abbonamento.

Non tutti gli abbonamenti danno accesso ai Release Day (essendoci anche l'abbonamento 'free' che non dà accesso).

2. *[Abbonamento, Catalogo]*

Molteplicità: 1 a N

Un abbonamento fa accedere ad N cataloghi.

Un catalogo viene acceduto da 1 abbonamento.

Totalità: Presente ad ambo i lati

Tutti gli abbonamenti fanno accedere a cataloghi.

Tutti i cataloghi sono acceduti da abbonamenti

3. *[Abbonamento, Cliente]*

Molteplicità: N a M

Un cliente può effettuare M abbonamenti.

Un abbonamento può essere effettuato da N clienti.

Totalità: Presente ad ambo i lati

Tutti i clienti effettuano un abbonamento (avendo sempre attivo anche l'abbonamento 'free').

Tutti gli abbonamenti sono effettuati da clienti.

4. *[Cliente, Carta]*

Molteplicità: 1 a N

Un cliente inserisce N carte.

Una carta è inserita da 1 cliente.

Totalità: Presente ad ambo i lati

Tutti i clienti inseriscono carte.

Tutte le carte sono inserite da un cliente.

5. *[Catalogo, Prodotto]*

Molteplicità: 1 a N

Un catalogo contiene N prodotti.

Un prodotto è contenuto in un catalogo.

Totalità: Presente ad ambo i lati

Tutti i cataloghi contengono prodotti.

Tutti i prodotti sono contenuti in un catalogo.

6. *[Carta, Acquisto]*

Molteplicità: N a 1

Una carta viene selezionata per N acquisti.

Un acquisto seleziona una carta.

Totalità: Presente solo dal lato di acquisto

Non tutte le carte vengono selezionate per un acquisto (un cliente può anche inserire una carta, ma non effettuare nessuno acquisto con quest'ultima).

Tutti gli acquisti vengono selezionati da carte.

7. *[Acquisto, Prodotto]*

Molteplicità: 1 a N

Un acquisto può contenere N prodotti.

Un prodotto è contenuto in 1 acquisto.

Totalità: Presente solo dal lato di acquisto

Tutti gli acquisti contengono prodotti.

Non tutti i prodotti sono contenuti in un acquisto.

8. *[Acquisto, Promo]*

Molteplicità: N a 1

Un acquisto applica 1 promo.

Una promo viene applicata ad N acquisti.

Totalità: Presente solo dal lato di promo

Non tutti gli acquisti applicano promo (un cliente può anche possedere una promo e non usarla o non possederne nessuna).

Tutte le promo sono applicate a degli acquisti.

9. *[Scarpe, Taglia scarpe]*

Molteplicità: M a N

Un paio di scarpe è suddiviso in N taglie.

Una taglia è stata suddivisa per M scarpe.

Totalità: Presente ad ambo i lati

Tutte le scarpe sono suddivise in taglie.

Tutte le taglie sono state suddivise per le scarpe.

10. *[Prodotto, Fornitura]*

Molteplicità: 1 a N

Un prodotto fa parte di N forniture.

Una fornitura è costituita da 1 prodotto.

Totalità: Presente ad ambo i lati

Tutti i prodotti fanno parte di forniture.

Tutte le forniture sono costituite da prodotti.

11. *[Fornitore, Fornitura]*

Molteplicità: 1 a N

Un fornitore spedisce N forniture.

Una fornitura è spedita da 1 fornitore.

Totalità: Presente solo dal lato di fornitura

Non tutti i fornitori spediscono una fornitura.

Tutte le forniture sono spedite da un fornitore.

12. *[Fornitura, Magazzino]*

Molteplicità: N a M

Una fornitura è contenuta in N magazzini.

Un magazzino contiene M forniture.

Totalità: Presente ad ambo i lati

Tutti i magazzini contengono forniture.

Tutte le forniture sono contenute in magazzini.

Associazioni N a M

Ad alcune associazioni N a M sono stati aggiunti, oltre alle chiavi, ulteriori attributi.

Effettua:

- Data inizio abbonamento: indica la data in cui il cliente sottoscrive un determinato tipo di abbonamento;
- Data fine abbonamento: indica la data in cui scade l'abbonamento sottoscritto dal cliente.

È contenuta in:

- Quantità distribuita: indica la quantità, associata ad una fornitura, che è stata distribuita in un determinato magazzino.

NUMERO MAGAZZINO	ID FORNITURA	QUANTITA' DISTRIBUITA
Mgz01	Frn01	40
Mgz02	Frn01	30

CAPITOLO 4

Prestazioni attese, numero potenziale utenti

Il sistema informatico in dotazione permette l'accesso al database ad un massimo di 20 categorie di utenti differenti. Le varie categorie non hanno limiti sul numero di persone di cui ne fanno parte, però c'è da precisare che il database è progettato per:

- Gestire massimo 1500 accessi quotidiani totali
- Gestire 500 accessi contemporaneamente
- Gestire massimo 45000 accessi mensili

Per superare questi limiti, i progettisti saranno costretti a modificare la struttura della base di dati.

Il tempo di esecuzione delle varie interrogazioni, viste, procedure, accessi al database è tale da non creare rallentamenti nell'uso della base di dati.

Solo se si forzasse il numero di utenti collegati alla base di dati contemporaneamente, potrebbero riscontrarsi piccoli problemi di rallentamento, ma è fortemente sconsigliato per i limiti indicati precedentemente.

Dimensione Tabelle

Il numero massimo di tabelle è stato definito dai progettisti durante la fase di progettazione del database. Se si vuole aumentare il numero di tabelle si dovrà cambiare la struttura del Database, e ciò influirà anche sulle prestazioni e sul numero potenziale di utenti della base di dati stessa.

Il database prevede 18 tabelle di cui:

- 11 sono entità normali;
- 2 sono tabelle dovute a specializzazioni;
- 4 sono tabelle dovute a molteplicità N a M;
- 1 è la tabella dovuta ad entità debole.

Successivamente le tabelle sopracitate sono state popolate con dei dati fittizi.

Di seguito il numero di tuple utilizzate per ogni tabella:

NOME TABELLA	TOTALE TUPLE
Cliente	15
Abbonamento	15
Catalogo	15
Release Day	16
Carta	15
Acquisto	15
Promo	15
Prodotto	30
Fornitura	15
Fornitore	15
Magazzino	15
Scarpe	15

Capi Hype	15
Taglia Scarpe	15
Da accesso	30
Effettua	27
Suddivise in	30
E contenuta in	15

Categorie Utenti

Le categorie d'utenti scelti sono:

- Cliente
- Gestore_Vendite
- Manager_Servizi
- Manager_Logistica
- Gestore_Prodotti

Ciò però non toglie che si possano aggiungere altre categorie d'utenti.

Tavola dei Volumi

La seguente tavola dei volumi è organizzata in base al volume che la base di dati potrebbe occupare in un anno.

Di seguito verrà riportata la tabella con i possibili dati all'interno:

NOME TABELLA	Volume
Cliente	90000
Abbonamento	15
Catalogo	90
Release Day	20
Carta	270000
Acquisto	360000
Promo	100
Prodotto	200
Fornitura	2000
Fornitore	15
Magazzino	15
Scarpe	100
Capi Hype	100
Taglia Scarpe	15
Da accesso	225000
Effettua	180000
Suddivise in	15000
E contenuta in	30000

Privilegi d'accesso

Cliente:

Non può in nessun modo alterare la base di dati, ma può visualizzare parti di esso, come: i suoi dati, le sue carte, i suoi acquisti, gli abbonamenti e i privilegi concessi dai vari abbonamenti.

Gestore Vendite:

Il Gestore Vendite si occupa principalmente della vendita dei prodotti. Egli è quindi responsabile della gestione degli acquisti del cliente e delle carte di credito da lui scelte.

Manager Servizi:

Il manager dei servizi è il responsabile della gestione degli abbonamenti effettuati dal cliente, e dei privilegi che ne conseguono, quindi anche della gestione dei cataloghi e dei Release Day.

Manager Logistica:

Il manager impiegato alla logistica è colui che si occupa della gestione delle forniture, quando vengono acquistate e quando devono essere smistate all'interno dei magazzini, e di mettersi in contatto con i fornitori (o di cambiarli nel caso in cui risultino inattivi).

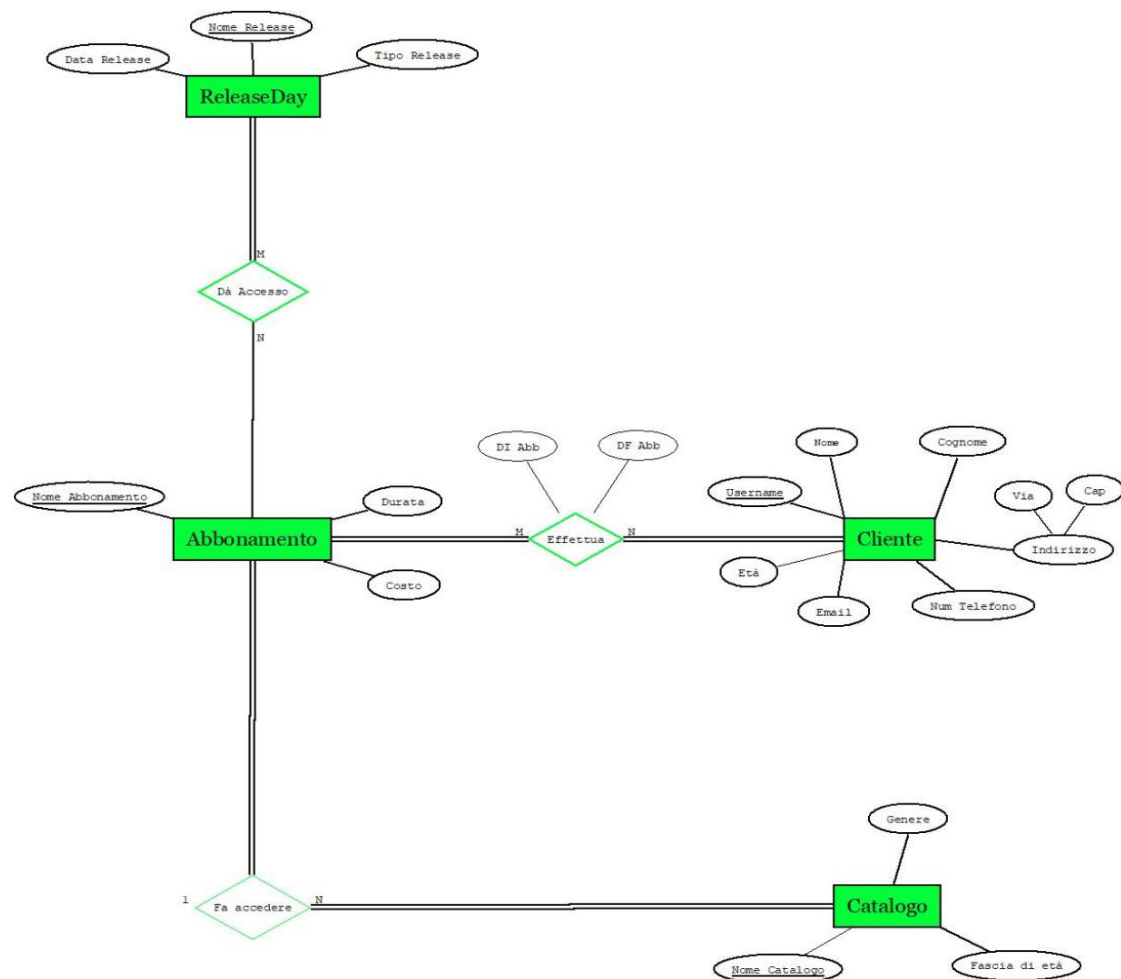
Gestore Prodotti:

Il gestore dei prodotti è colui che si occupa di gestire i vari tipi di prodotti, gestendone anche la variazione di prezzo nel corso del tempo.

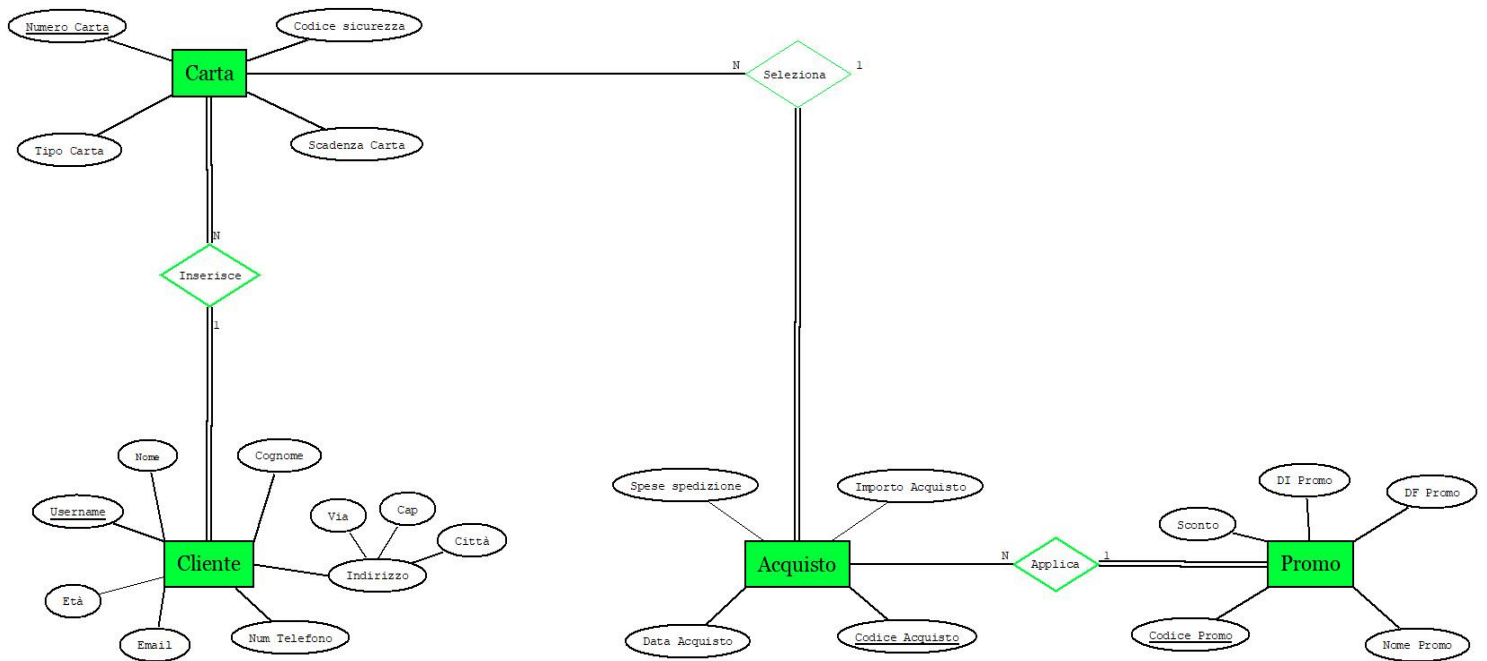
CAPITOLO 5

Schemi scheletri di settore

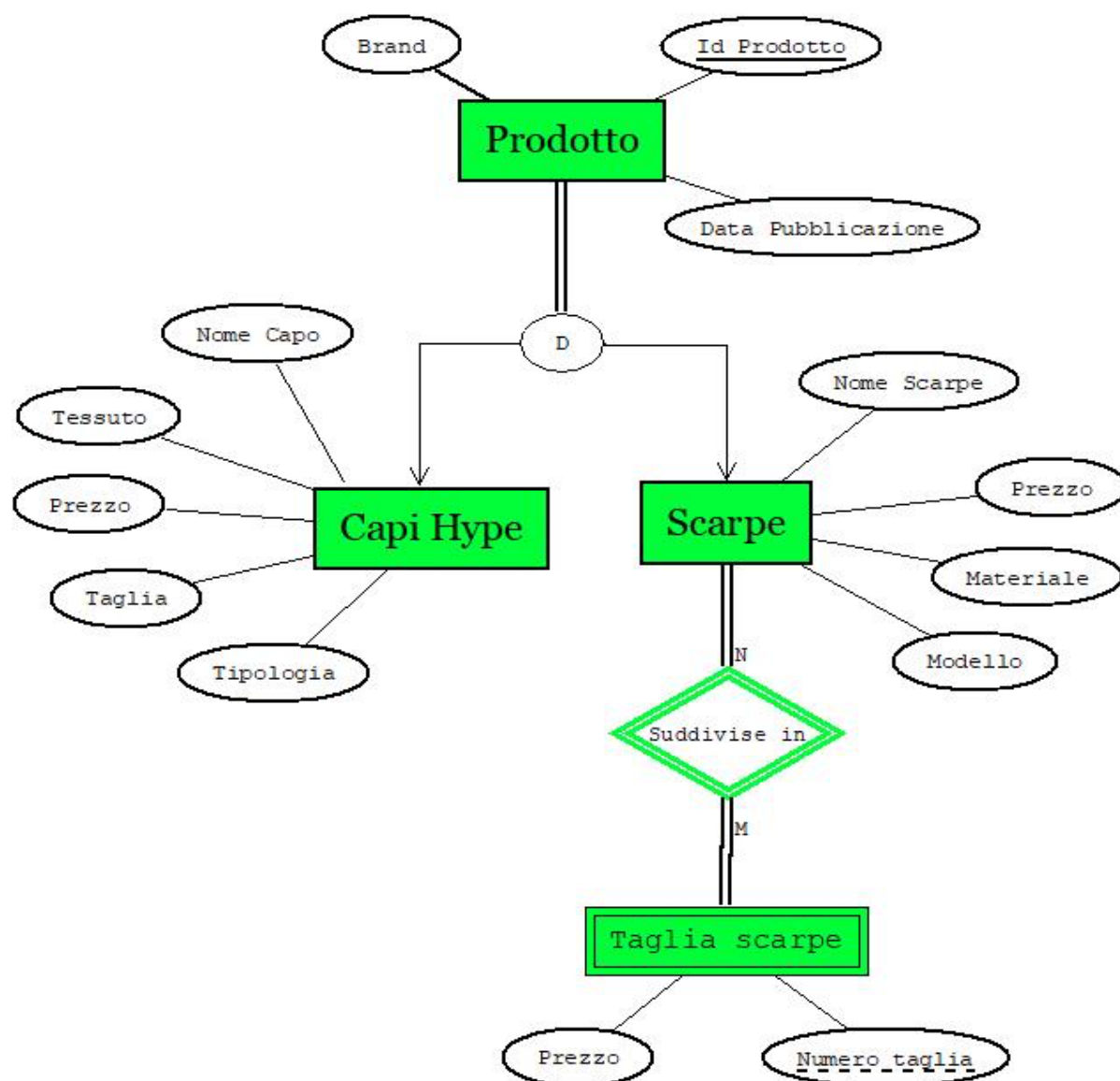
1. Scheletro di settore della macro-area Servizi:



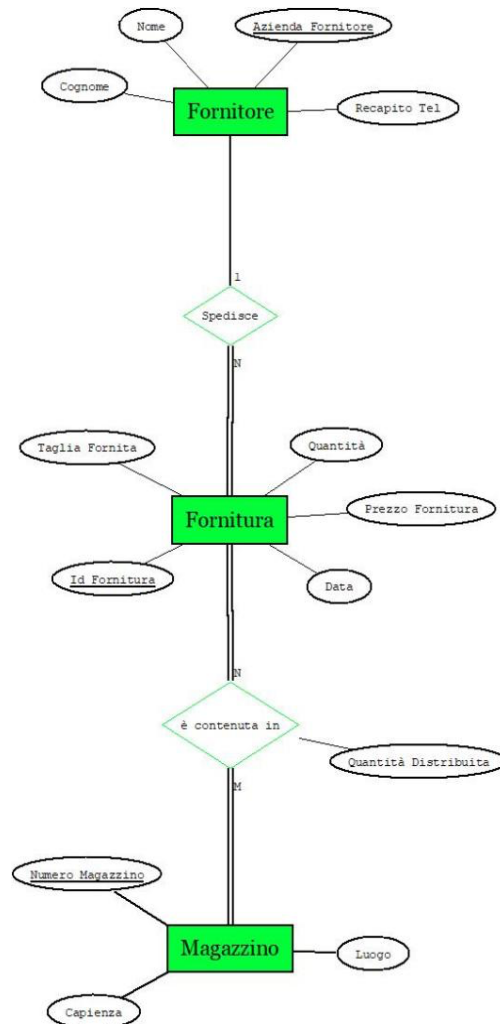
2. Scheletro di settore della macro-area Acquisto:



3. *Scheletro di settore della macro-area Prodotti:*



4. Scheletro di settore della macro-area Logistica:

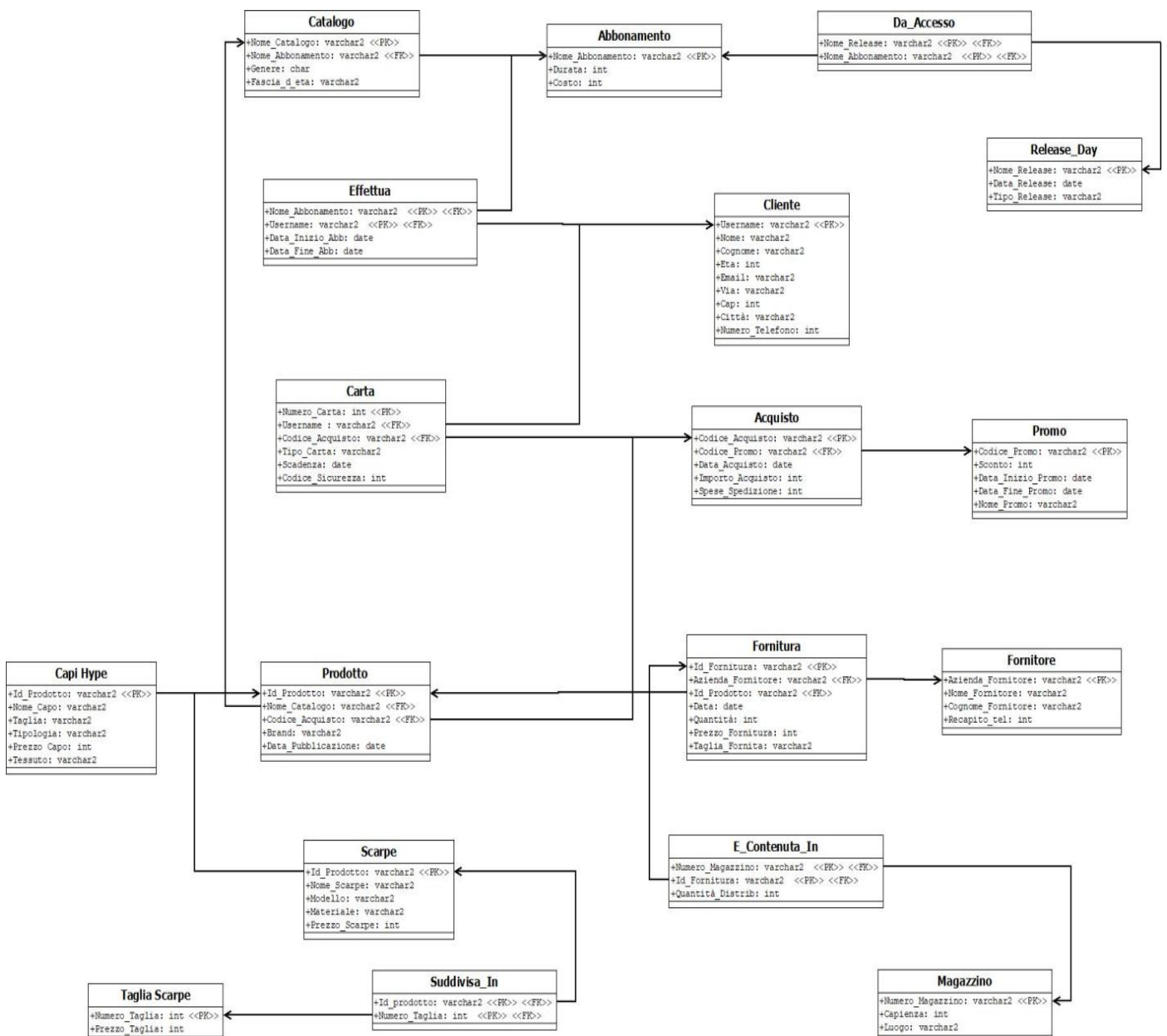


Rappresentazione grafica diagramma EE/R

Per una visione più comprensibile del diagramma visionare i file .jpg (medesimo consiglio utilizzabile per gli altri diagrammi).

CAPITOLO 6

Diagramma relazionale



CAPITOLO 7

Elenco delle operazioni

Di seguito verranno elencate le operazioni di base e le operazioni degli utenti, effettuabili sulla base di dati.

Operazioni di base

1. Cliente:
 - Aggiornamento numero di telefono
 - Aggiornamento e-mail
 - Cancellazione di un cliente
 - Aggiornamento Indirizzo

2. Abbonamento:
 - Aggiornamento durata
 - Inserimento nuovo tipo di abbonamento
 - Cancellazione di un abbonamento

3. Release Day:
 - Inserimento nuova release day
 - Cancellazione release day

4. Catalogo:
 - Aggiornamento di un catalogo
 - Inserimento nuovo catalogo

5. Carta:
 - Inserimento di una nuova carta
 - Cancellazione di una carta

6. Acquisto:
 - Aggiornamento dell'importo acquisto
 - Inserimento di un nuovo acquisto

7. Promo:
 - Inserimento di una nuova promo
 - Cancellazione di una promo

8. Prodotti:
 - Inserimento di un nuovo prodotto
 - Cancellazione di un prodotto

9. Scarpe:
-Aggiornamento del prezzo di una scarpa
-Inserimento di una nuova scarpa
-Cancellazione di una scarpa
10. Capi Hype:
-Aggiornamento del prezzo dei capi
-Inserimento di un nuovo cap
-Cancellazione di un capo
11. Fornitura:
-Inserimento nuova fornitura
12. Fornitore:
-Inserimento nuovo fornitore
-Cancellazione fornitore
-Aggiornamento recapito telefonico fornitore
13. Magazzino:
-Aggiornamento capienza magazzino

Operazioni degli utenti

Di seguito verranno elencati, alcuni esempi, di operazioni eseguibili da parte degli utenti:

Cliente:

- Visualizzare l'età massima dei clienti della base di dati
- Visualizzare l'età minima dei clienti della base di dati

Gestore Vendite:

- Visualizzare le promo programmate, o utilizzabili, da un determinato giorno in poi
- Visualizzare gli acquisti effettuati durante l'anno

Gestore Prodotti:

- Visualizzare tutte le sneakers presenti nel database
- Visualizzare i capi con prezzo superiore a 150€

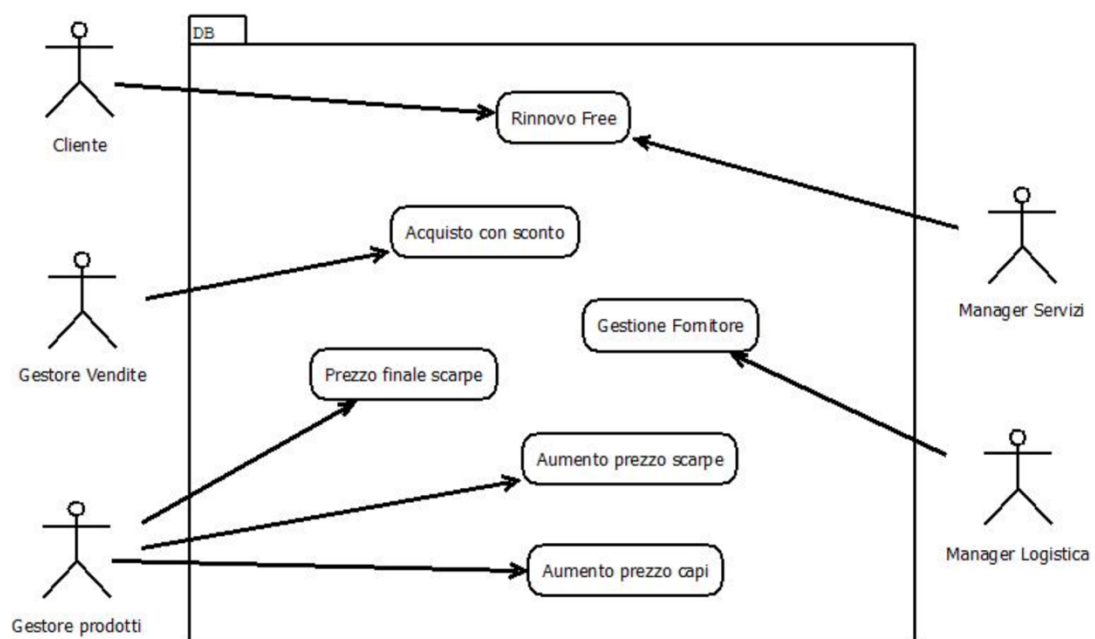
Manager servizi:

- Visualizzare i cataloghi con una determinata fascia d'età
- Visualizzare i Release Day rimasti

Manager Logistica:

- Visualizzare i fornitori che hanno spedito forniture negli ultimi 3 mesi
- Visualizzare la fornitura con il prezzo più alto
- Visualizzare la fornitura con il prezzo più basso

Diagramma Uml Use Case



CAPITOLO 8

Vincoli d'integrità statici

1. *L'attributo e-mail deve assumere determinati valori al proprio interno:*
`EMAIL_MASK CHECK (`
`REGEXP_LIKE (email, '^\\w+.*@{1}\\w+.*$'))`
2. *L'attributo Data_inizio_abb deve sempre essere minore o uguale alla Data_fine_abb:*
`CHECK (DATA_INIZIO_ABB <= DATA_FINE_ABB)`
3. *L'attributo Data_inizio_promo deve sempre essere minore o uguale alla Data_fine_promo:*
`CHECK (DATA_INIZIO_PROMO <=`
`DATA_FINE_PROMO)`
4. *Inoltre, la durata in giorni tra la Data_inizio_promo e la Data_fine_promo non deve mai superare i 10 giorni:*
`CHECK ((DATA_FINE_PROMO -`
`DATA_INIZIO_PROMO) <= 10)`
5. *L'attributo Importo_acquisto non deve mai superare i 12000 euro, impostato come tetto massimo d'acquisto:*

*IMPORTO_ACQUISTO NUMBER CHECK
(IMPORTO_ACQUISTO IS NOT NULL AND
IMPORTO_ACQUISTO < 12000)*

6. *L'attributo Spese_spedizione deve avere un importo compreso tra:
SPESE_SPEDIZIONE NUMBER CHECK
(SPESE_SPEDIZIONE BETWEEN 10 AND 20)*

7. *L'attributo genere deve essere uguale a ('M', 'F', 'U'). Queste lettere indicano rispettivamente: maschile, femminile, unisex:
GENERE CHAR(1) CHECK (GENERE IN ('M','F','U'))*

8. *L'attributo taglia, in capi hype, deve corrispondere esattamente ad una delle seguenti taglie:
TAGLIA VARCHAR2(4) CHECK (TAGLIA IN ('XS','S','M','L','XL','XXL'))*

9. *L'attributo numero_taglia deve avere una taglia compresa tra:
CHECK (NUMERO_TAGLIA BETWEEN 36 AND 51)*

10. *L'attributo Prezzo_fornitura deve essere minore a 15000 euro, tetto massimo di spesa impostato:
PREZZO_FORNITURA NUMBER CHECK
(PREZZO_FORNITURA IS NOT NULL AND
PREZZO_FORNITURA < 15000)*

Vincoli d'integrità dinamici

1. *Si vuole gestire il caso in cui due abbonamenti, ad eccezione del free, abbiano lo stesso nome ed hanno stesse date di inizio e di fine. Ciò viene fatto per evitare che un cliente effettui lo stesso abbonamento più volte, nello stesso lasso di tempo.*
2. *Si vuole gestire il caso in cui una carta, inserita successivamente all'interno del database, è scaduta.*
3. *Si vuole gestire il caso di un reso: se un acquisto contenuto all'interno del sito ha una data d'acquisto superiore a 31 giorni rispetto alla data di sistema, allora non sarà possibile eseguire il reso.*
4. *Si vuole gestire il caso in cui un prodotto deve essere contenuto all'interno di un solo catalogo.*
5. *Si vuole gestire il caso in cui vengano rilasciate un totale di Release Day in un anno superiori a quelle che sono quelle imposte da politica aziendale (in questo caso massimo 20 in un anno).*

6. *Si vuole gestire il caso in cui la somma totale delle quantità di prodotti distribuite all'interno di un magazzino superino la capienza del magazzino in cui si trovano*

Vincoli d'integrità referenziale

1. *[Dà accesso -> Release Day]
-Chiave esterna: Nome Release
-Motivazione: La chiave esterna si trova in Da_accesso in quanto tra le 2 entità c'è una relazione N a M*

2. *[Dà accesso -> Abbonamento]
-Chiave esterna: Nome Abbonamento
-Motivazione: La chiave esterna si trova in Da_accesso in quanto tra le 2 entità c'è una relazione N a M*

3. *[Effettua -> Abbonamento]*
 - Chiave esterna: Nome Abbonamento
 - Motivazione: La chiave esterna si trova in Effettua in quanto tra le 2 entità c'è una relazione N a M

4. *[Catalogo -> Abbonamento]*
 - Chiave esterna: Nome Abbonamento
 - Motivazione: La chiave esterna si trova in Catalogo in quanto tra le 2 entità c'è una relazione N a 1

5. *[Effettua -> Cliente]*
 - Chiave esterna: Username
 - Motivazione: La chiave esterna si trova in Effettua in quanto tra le 2 entità c'è una relazione N a M

6. *[Prodotto -> Catalogo]*
 - Chiave esterna: Nome Catalogo
 - Motivazione: La chiave esterna si trova in Prodotto in quanto tra le 2 entità c'è una relazione N a 1

7. *[Carta -> Cliente]*
 - Chiave esterna: Username
 - Motivazione: La chiave esterna si trova in Carta in quanto tra le 2 entità c'è una relazione N a 1

8. [Carta -> Acquisto]
 - Chiave esterna: Codice Acquisto
 - Motivazione: La chiave esterna si trova in Carta in quanto tra le 2 entità c'è una relazione N a 1

9. [Acquisto -> Promo]
 - Chiave esterna: Codice Promo
 - Motivazione: La chiave esterna si trova in Acquisto in quanto tra le 2 entità c'è una relazione N a 1

10. [Prodotto -> Acquisto]
 - Chiave esterna: Codice Acquisto
 - Motivazione: La chiave esterna si trova in Prodotto in quanto tra le 2 entità c'è una relazione N a 1

11. [Scarpe -> Prodotto]
 - Chiave esterna: Id Prodotto
 - Motivazione: È stata creata una nuova tabella per l'entità Scarpe poiché essa è una sottoclasse dell'entità Prodotto. La nuova tabella avrà come chiave esterna la chiave primaria della superclasse (da notare che essa sarà comune per tutte le sottoclassi, quindi anche per Capi Hype)

12. [Capi Hype -> Prodotto]
-Chiave esterna: Id Prodotto
-Motivazione: È stata creata una nuova tabella per l'entità Capi hype poiché essa è una sottoclasse dell'entità Prodotto. La nuova tabella avrà come chiave esterna la chiave primaria della superclasse (da notare che essa sarà comune per tutte le sottoclassi, quindi anche per Scarpe)
13. [Suddivise In -> Scarpe]
-Chiave esterna: Id Prodotto
-Motivazione: La chiave esterna si trova in Suddivise_in in quanto tra le 2 entità c'è una relazione N a M
14. [Suddivise in -> Taglia Scarpe]
-Chiave esterna: Numero Taglia
-Motivazione: La chiave esterna si trova in Suddivise_in in quanto tra le 2 entità c'è una relazione N a M
15. [Fornitura -> Prodotto]
-Chiave esterna: Id Prodotto
-Motivazione: La chiave esterna si trova in Fornitura in quanto tra le 2 entità c'è una relazione N a 1

16. *[Fornitura -> Fornitore]*
-Chiave esterna: Azienda fornitore
-Motivazione: La chiave esterna si trova in Fornitura in quanto tra le 2 entità c'è una relazione N a 1
17. *[È contenuta in -> Fornitura]*
-Chiave esterna: Id Fornitura
-Motivazione: La chiave esterna si trova in E_contenuta_in in quanto tra le 2 entità c'è una relazione N a M
18. *[È contenuta in -> Magazzino]*
-Chiave esterna: Numero Magazzino
-Motivazione: La chiave esterna si trova in E_contenuta_in in quanto tra le 2 entità c'è una relazione N a M

CAPITOLO 9

Normalizzazione

In seguito alla verifica di normalità dello schema non si sono verificate dipendenze anomale.

Per la verifica della prima forma normale, le relazioni non devono contenere attributi multivalore o composti, ed in effetti è risultato che essi sono assenti.

Per la verifica della seconda forma normale, le relazioni in cui la chiave primaria è costituita da più attributi, gli attributi non chiave non devono dipendere funzionalmente solo da una parte della chiave primaria, ed in effetti è risultato che non vi sono chiavi primarie composte da due attributi.

Per la verifica della terza forma normale, le relazioni non devono contenere attributi non chiave funzionalmente dipendenti da altri attributi non chiave (o da un insieme di attributi non chiave), ed in effetti non esiste alcuna dipendenza transitiva tra attributi non chiave e la chiave primaria.

Infine, lo schema è conforme anche alla forma normale di Boys e Codd.

CAPITOLO 10

Creazione Tabelle

Di seguito verrà riportata la fase di creazione delle tabelle:

```
--TABLE CLIENTE
CREATE TABLE CLIENTE (
  USERNAME VARCHAR2(18) CONSTRAINT PK_CLIENTE PRIMARY KEY,
  NOME VARCHAR2(16) NOT NULL,
  COGNOME VARCHAR2(16) NOT NULL,
  ETA NUMBER CHECK (ETA > 17),
  EMAIL VARCHAR2(20) UNIQUE,
  CITTA VARCHAR2(14) NOT NULL,
  VIA VARCHAR2(20) NOT NULL,
  CAP NUMBER(5),
  NUMERO_TELEFONO NUMBER UNIQUE,
  CONSTRAINT EMAIL_MASK CHECK (
    REGEXP_LIKE(email, '^\\w+.*@{1}\\w+.*$')) -- mail@gmail.com
);
```

```
--TABLE ABBONAMENTO
CREATE TABLE ABBONAMENTO (
  NOME_ABBONAMENTO VARCHAR2(20) CONSTRAINT PK_ABBONAMENTO
  PRIMARY KEY,
  DURATA NUMBER(2), -- IN MESI
  COSTO NUMBER(3)
);
```

```
--TABLE EFFETTUA
CREATE TABLE EFFETTUA (
NOME_ABBONAMENTO VARCHAR2(20) NOT NULL,
USERNAME VARCHAR2(18) NOT NULL,
DATA_INIZIO_ABB DATE,
DATA_FINE_ABB DATE,
CHECK (DATA_INIZIO_ABB <= DATA_FINE_ABB),
PRIMARY KEY(NOME_ABBONAMENTO,USERNAME),
CONSTRAINT FK_EFFETTUA_ABB FOREIGN KEY (NOME_ABBONAMENTO)
REFERENCES ABBONAMENTO(NOME_ABBONAMENTO),
CONSTRAINT FK_EFFETTUA_USER FOREIGN KEY (USERNAME) REFERENCES
CLIENTE(USERNAME)
);
```

```
--TABLE RELEASE DAY
CREATE TABLE RELEASE_DAY (
NOME_RELEASE VARCHAR2(20) CONSTRAINT PK_RELEASE_DAY PRIMARY
KEY,
DATA_RELEASE DATE UNIQUE,
TIPO_RELEASE VARCHAR2(30) NOT NULL
);
```

```
--TABLE DA ACCESSO
CREATE TABLE DA_ACCESSO (
NOME_RELEASE VARCHAR2(20) NOT NULL,
NOME_ABBONAMENTO VARCHAR2(20) NOT NULL,
PRIMARY KEY(NOME_RELEASE,NOME_ABBONAMENTO),
CONSTRAINT FK_ACCESSO_REL FOREIGN KEY (NOME_RELEASE)
REFERENCES RELEASE_DAY(NOME_RELEASE),
CONSTRAINT FK_ACCESSO_ABB FOREIGN KEY (NOME_ABBONAMENTO)
REFERENCES ABBONAMENTO(NOME_ABBONAMENTO)
);
```

```
--TABLE PROMO
CREATE TABLE PROMO (
CODICE_PROMO VARCHAR2(20) CONSTRAINT PK_PROMO PRIMARY KEY,
SCONTO NUMBER DEFAULT 0,
DATA_INIZIO_PROMO DATE,
DATA_FINE_PROMO DATE,
CHECK (DATA_INIZIO_PROMO <= DATA_FINE_PROMO),
--OLTRE AL CHECK PER IL CONTROLLO DELLA DATA, INSERITO ANCHE
QUELLO DELLA DURATA
--CHE, ESSENDO VARIABILE, NON DEVE MAI SUPERARE I 10 GIORNI
CHECK ((DATA_FINE_PROMO - DATA_INIZIO_PROMO) <= 10),
NOME_PROMO VARCHAR2(20) NOT NULL
);
```

```
--TABLE ACQUISTO
CREATE TABLE ACQUISTO (
CODICE_ACQUISTO VARCHAR2(20) CONSTRAINT PK_CODICE_ACQUISTO
PRIMARY KEY,
CODICE_PROMO VARCHAR2(20) NOT NULL,
DATA_ACQUISTO DATE NOT NULL,
IMPORTO_ACQUISTO NUMBER CHECK (IMPORTO_ACQUISTO IS NOT NULL
AND IMPORTO_ACQUISTO < 12000), --TETTO MASSIMO D'ACQUISTO
SPESE_SPEDIZIONE NUMBER CHECK (SPESE_SPEDIZIONE BETWEEN 10 AND
20),
CONSTRAINT FK_PROMO_ACQUISTO FOREIGN KEY (CODICE_PROMO)
REFERENCES PROMO(CODICE_PROMO)
);
```

```
--TABLE CATALOGO
CREATE TABLE CATALOGO (
NOME_CATALOGO VARCHAR2(16) CONSTRAINT PK_CATALOGO PRIMARY KEY,
NOME_ABBONAMENTO VARCHAR2(20) NOT NULL,
GENERE CHAR(1) CHECK (GENERE IN ('M', 'F', 'U')), --
UOMO, DONNA, UNISEX
FASCIA_D_ETA VARCHAR2(16) NOT NULL,
CONSTRAINT FK_CATALOGO FOREIGN KEY (NOME_ABBONAMENTO)
REFERENCES ABBONAMENTO(NOME_ABBONAMENTO)
);
```

```
--TABLE CARTA
CREATE TABLE CARTA (
NUMERO_CARTA NUMBER(8) CONSTRAINT PK_CARTA PRIMARY KEY,
USERNAME VARCHAR2(18) NOT NULL,
CODICE_ACQUISTO VARCHAR2(20) NOT NULL,
TIPO_CARTA VARCHAR2(14) NOT NULL,
SCADENZA DATE NOT NULL,
CODICE_SICUREZZA NUMBER(3),
CONSTRAINT FK_CARTA_USER FOREIGN KEY (USERNAME) REFERENCES
CLIENTE(USERNAME),
CONSTRAINT FK_CARTA_ACQUISTO FOREIGN KEY (CODICE_ACQUISTO)
REFERENCES ACQUISTO(CODICE_ACQUISTO)
);
```

```
--TABLE MAGAZZINO
CREATE TABLE MAGAZZINO (
NUMERO_MAGAZZINO VARCHAR2(5) CONSTRAINT PK_MAGAZZINO PRIMARY
KEY,
CAPIENZA NUMBER CHECK (CAPIENZA < 301),
LUOGO VARCHAR2(14) NOT NULL
);
```

```
--TABLE PRODOTTO
CREATE TABLE PRODOTTO (
ID_PRODOTTO VARCHAR2(20) CONSTRAINT PK_PRODOTTO PRIMARY KEY,
NOME_CATALOGO VARCHAR2(16) NOT NULL,
CODICE_ACQUISTO VARCHAR2(20) NOT NULL,
BRAND VARCHAR2(16) NOT NULL,
DATA_PUBBLICAZIONE DATE, --PUBBLICAZIONE SUL SITO
CONSTRAINT FK_PRODOTTO_CATALOGO FOREIGN KEY (NOME_CATALOGO)
REFERENCES CATALOGO(NOME_CATALOGO),
CONSTRAINT FK_PRODOTTO_ACQUISTO FOREIGN KEY (CODICE_ACQUISTO)
REFERENCES ACQUISTO(CODICE_ACQUISTO)
);
```

```
--TABLE CAPI HYPE
CREATE TABLE CAPI_HYPE (
ID_PRODOTTO VARCHAR2(20) CONSTRAINT PK_CAPI_HYPE PRIMARY KEY,
NOME_CAPO VARCHAR2(20) NOT NULL,
TAGLIA VARCHAR2(4) CHECK (TAGLIA IN
('XS','S','M','L','XL','XXL')),
PREZZO_CAPO NUMBER(4),
TIPOLOGIA VARCHAR2(16) NOT NULL,
TESSUTO VARCHAR2(16) NOT NULL,
CONSTRAINT FK_CAPI FOREIGN KEY (ID_PRODOTTO) REFERENCES
PRODOTTO(ID_PRODOTTO)
);
```

```
--TABLE SCARPE
CREATE TABLE SCARPE (
ID_PRODOTTO VARCHAR2(20) CONSTRAINT PK_SCARPE PRIMARY KEY,
NOME_SCARPE VARCHAR2(26) NOT NULL,
MODELLO VARCHAR2(16) NOT NULL,
MATERIALE VARCHAR2(16) NOT NULL,
PREZZO_SCARPE NUMBER(4),
CONSTRAINT FK_SCARPE FOREIGN KEY (ID_PRODOTTO) REFERENCES
PRODOTTO(ID_PRODOTTO)
);
```

```
--TABLE TAGLIA SCARPE
CREATE TABLE TAGLIA_SCARPE (
NUMERO_TAGLIA NUMBER(2) CONSTRAINT PK_TAGLIA_SCARPE PRIMARY
KEY,
CHECK (NUMERO_TAGLIA BETWEEN 36 AND 51),
PREZZO_TAGLIA NUMBER(4)
);
```



```
--TABLE SUDDIVISE IN
CREATE TABLE SUDDIVISE_IN (
ID_PRODOTTO VARCHAR2(20),
NUMERO_TAGLIA NUMBER(2),
PRIMARY KEY(ID_PRODOTTO,NUMERO_TAGLIA),
CONSTRAINT FK_SUDDIVIS_SCARPE FOREIGN KEY (ID_PRODOTTO)
REFERENCES SCARPE(ID_PRODOTTO),
CONSTRAINT FK_SUDDIVIS_TAGLIA FOREIGN KEY (NUMERO_TAGLIA)
REFERENCES TAGLIA_SCARPE(NUMERO_TAGLIA)
);
```

```
--TABLE FORNITORE
CREATE TABLE FORNITORE (
AZIENDA_FORNITORE VARCHAR2(18) CONSTRAINT PK_FORNITORE PRIMARY
KEY,
NOME_FORNITORE VARCHAR2(16) NOT NULL,
COGNOME_FORNITORE VARCHAR2(16) NOT NULL,
RECAPITO_TEL NUMBER UNIQUE
);
```

```
--TABLE FORNITURA
CREATE TABLE FORNITURA (
ID_FORNITURA VARCHAR2(5) CONSTRAINT PK_FORNITURA PRIMARY KEY,
AZIENDA_FORNITORE VARCHAR2(18) NOT NULL,
ID_PRODOTTO VARCHAR2(20) NOT NULL,
DATA_FORNITURA DATE NOT NULL,
QUANTITA NUMBER(3),
TAGLIA_FORNITA VARCHAR2(5) NOT NULL,
PREZZO_FORNITURA NUMBER CHECK (PREZZO_FORNITURA IS NOT NULL
AND PREZZO_FORNITURA < 15000), --TETTO SPESA MASSIMO
CONSTRAINT FK_AZIENDA_FORNITA FOREIGN KEY (AZIENDA_FORNITORE)
REFERENCES FORNITORE(AZIENDA_FORNITORE),
CONSTRAINT FK_PRODOTTO_FORNITA FOREIGN KEY (ID_PRODOTTO)
REFERENCES PRODOTTO(ID_PRODOTTO)
);
```

```
--TABLE E' CONTENUTA IN
CREATE TABLE E_CONTENUTA_IN (
NUMERO_MAGAZZINO VARCHAR2(5),
ID_FORNITURA VARCHAR2(5),
QUANTITA_DISTRIB NUMBER(3), --QUANTITA' PRECISA DI UNA SINGOLA
FORNITURA IN UN MAGAZZINO
PRIMARY KEY(NUMERO_MAGAZZINO,ID_FORNITURA),
CONSTRAINT FK_CONTIENE_MAGAZZINO FOREIGN KEY
(NUMERO_MAGAZZINO) REFERENCES MAGAZZINO(NUMERO_MAGAZZINO),
CONSTRAINT FK_CONTIENE_FORNITURA FOREIGN KEY (ID_FORNITURA)
REFERENCES FORNITURA(ID_FORNITURA)
);
```

Popolamento

Ogni tabella verrà popolata con dei dati fittizi. Di seguito verrà quindi riportato il popolamento:

```
--ORDINE POPOLAMENTO:
--FORNITORE
--MAGAZZINO
--ABBONAMENTO
--RELEASE_DAY
--DA_ACCESSO
--PROMO
--ACQUISTO
--CLIENTE
--EFFETTUA
--CARTA
--CATALOGO
--PRODOTTO
--CAPI_HYPE
--SCARPE
--TAGLIA_SCARPE
--SUDDIVISE_IN
--FORNITURA
--E_CONTENUTA_IN
```

```

--FORNITORE
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('YEEZY','KANYE','WEST',3334445665);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES
('JORDAN','VINCENZO','SALZANO',3334445667);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('OFF_WHITE','VIRGIL','ABLOH',3334445678);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('VLONE','GENNARO','FORTE',3334445443);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES
('CACTUS_JACK','TRAVIS','SCOTT',3334445699);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('MCQUEEN','ALEX','FERRI',3334445661);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('SUPREME','DAVIDE','SOPRANO',3334445621);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('BAPE','SIMONE','D_ANGELO',3334445663);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES
('THE_NORTH_FACE','RAFFAELE','SPADA',3334445543);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('DIOR','PATRIZIO','VESTE',3334445655);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('AMIRI','DARIO','SIGNORE',3334445668);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES
('GIVENCHY','GIOVANNI','VENEZIA',3334445656);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('OVO','DINO','BRASCHI',3334445555);
INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('NIKE','NICOLA','NOTE',3334445999);

```

```

INSERT INTO
FORNITORE(AZIENDA_FORNITORE,NOME_FORNITORE,COGNOME_FORNITORE,R
ECAPITO_TEL) VALUES ('ADIDAS','ANDREA','ANTONINI',3334445688);

```

--MAGAZZINO

```

INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ01',300,'NAPOLI');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ02',300,'MILANO');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ03',300,'ROMA');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ04',300,'CATANIA');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ05',300,'TORINO');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ06',300,'CAGLIARI');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ07',300,'BERGAMO');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ08',300,'TRENTO');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ09',300,'VARESE');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ10',300,'BARI');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ11',300,'SALERNO');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ12',300,'VENEZIA');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ13',300,'GENOVA');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ14',300,'BOLOGNA');
INSERT INTO MAGAZZINO(NUMERO_MAGAZZINO,CAPIENZA,LUOGO) VALUES
('MGZ15',300,'FIRENZE');

```

--ABBONAMENTO

```

INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('FREE',36, 0);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('WOOD',1, 3);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('BRICK',2, 5);

```

```

INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('GLASS',3, 7);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('PLASTIC',4, 10);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('BRONZE',5, 12);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('STONE',6, 15);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('IRON',7, 18);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('MERCURY',8, 20);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('NICKEL',9, 22);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('STEEL',10, 25);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('SILVER',11, 32);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('GOLD',12, 28);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('DIAMOND',15, 37);
INSERT INTO ABBONAMENTO(NOME_ABBONAMENTO,DURATA,COSTO) VALUES
('PLATINUM',24, 50);

```

```

--RELEASE_DAY
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('ADIDAS_RELEASE',DATE'2022-01-01', 'YEEZY_350V2');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('NIKE_RELEASE',DATE'2022-02-01', 'AIRFORCE_1');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('OFF_WHITE_RELEASE',DATE'2022-03-01',
'JORDAN_4_RETRO_OFF_WHITE_SAIL');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('SPRING_RELEASE',DATE'2022-03-21', 'AIRJORDAN_HIGH');

```

```

INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('VLONE_RELEASE',DATE'2022-04-01',
'JUICE_WRLD_X_VLONE_999_T_SHIRT');

```

```

INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('PALACE_RELEASE',DATE'2022-05-01', 'PALACE_P-3-K-9_T');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('SUPREME_RELEASE',DATE'2022-06-01', 'SUPREME_BOX_LOGO');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('SUMMER_RELEASE',DATE'2022-06-21', 'CROCS_POLLEX');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('JORDAN_RELEASE',DATE'2022-07-01', 'JORDAN_1_OG');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('OVO_RELEASE',DATE'2022-08-01', 'DRAKE_SNEAKERS');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('CONVERSE_RELEASE',DATE'2022-09-01',
'CONVERSE_CHUCK_TAYLOR_ALL-STAR');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('AUTUMN_RELEASE',DATE'2022-09-21', 'ALEXANDER_MCQUEEN');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('NEWBALANCE_RELEASE',DATE'2022-10-01', 'NEW_BALANCE_550');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('BALENCIAGA_RELEASE',DATE'2022-11-01',
'BALENCIAGA_TRIPLE_S');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('CACTUS_JACK_RELEASE',DATE'2022-12-01',
'TRAVIS_SCOTT_CACTUS_JACK');
INSERT INTO
RELEASE_DAY(NOME_RELEASE,DATA_RELEASE,TIPO_RELEASE) VALUES
('WINTER_RELEASE',DATE'2022-12-21', 'NORTH_FACE_JACKET');

```

```

--DA ACCESSO
--OVVIAMENTE OGNI RELEASE E' ASSOCIATA A TUTTI GLI ABBONAMENTI
TRANNE QUELLO FREE,DI SEGUITO
--VENGONO ELENcate SOLO 2 RELEASE PER NON RENDERE IL
POPOLAMENTO TROPPO LUNGO
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','WOOD');

```

```

INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','BRICK');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','GLASS');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','PLASTIC');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','BRONZE');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','STONE');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','IRON');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','MERCURY');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','NICKEL');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','STEEL');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','GOLD');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','SILVER');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','DIAMOND');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('JORDAN_RELEASE','PLATINUM');

```

```

INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','WOOD');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','BRICK');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','GLASS');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','PLASTIC');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','BRONZE');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','STONE');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','IRON');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','MERCURY');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','NICKEL');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','STEEL');

```

```

INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','GOLD');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','SILVER');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','DIAMOND');
INSERT INTO DA_ACCESSO(NOME_RELEASE,NOME_ABBONAMENTO) VALUES
('CACTUS_JACK_RELEASE','PLATINUM');

```

--PROMO

```

INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM01',DATE'2022-03-14',DATE'2022-03-
21',2,'SCONT01');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM02',DATE'2022-02-24',DATE'2022-02-
28',2,'SCONT02');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM03',DATE'2022-03-04',DATE'2022-03-
14',2,'SCONT03');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM04',DATE'2022-05-14',DATE'2022-05-
21',2,'SCONT04');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM05',DATE'2022-02-23',DATE'2022-02-
27',2,'SCONT05');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM06',DATE'2022-05-15',DATE'2022-05-
20',3,'SCONT06');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM07',DATE'2022-04-23',DATE'2022-04-
30',3,'SCONT07');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM08',DATE'2022-02-04',DATE'2022-02-
13',3,'SCONT08');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NO
ME_PROMO) VALUES ('PRM09',DATE'2022-03-13',DATE'2022-03-
20',4,'SCONT09');

```



```

INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NOME_PROMO) VALUES ('PRM10',DATE'2022-02-14',DATE'2022-02-21',4,'SCONTO10');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NOME_PROMO) VALUES ('PRM11',DATE'2022-03-05',DATE'2022-03-13',5,'SCONTO11');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NOME_PROMO) VALUES ('PRM12',DATE'2022-01-05',DATE'2022-01-10',5,'SCONTO12');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NOME_PROMO) VALUES ('PRM13',DATE'2022-04-08',DATE'2022-04-12',5,'SCONTO13');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NOME_PROMO) VALUES ('PRM14',DATE'2022-02-07',DATE'2022-02-10',5,'SCONTO14');
INSERT INTO
PROMO(CODICE_PROMO,DATA_INIZIO_PROMO,DATA_FINE_PROMO,SCONTO,NOME_PROMO) VALUES ('PRM15',DATE'2022-03-12',DATE'2022-03-18',5,'SCONTO15');

```

--ACQUISTO

```

INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_ACQUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ01','PRM01',DATE'2022-03-24',450,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_ACQUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ02','PRM02',DATE'2022-03-10',830,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_ACQUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ03','PRM03',DATE'2022-05-14',180,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_ACQUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ04','PRM04',DATE'2022-05-15',150,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC

```

```

QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ05','PRM05',DATE'2022-03-
23',430,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ06','PRM06',DATE'2022-05-
17',1030,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ07','PRM07',DATE'2022-06-
16',970,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ08','PRM08',DATE'2022-02-
15',620,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ09','PRM09',DATE'2022-03-
21',970,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ10','PRM10',DATE'2022-03-
14',200,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ11','PRM11',DATE'2022-03-
15',270,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ12','PRM12',DATE'2022-01-
18',130,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ13','PRM13',DATE'2022-05-
08',240,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ14','PRM14',DATE'2022-02-
17',220,15);
INSERT INTO
ACQUISTO(CODICE_ACQUISTO,CODICE_PROMO,DATA_ACQUISTO,IMPORTO_AC
QUISTO,SPESE_SPEDIZIONE) VALUES ('ACQ15','PRM15',DATE'2022-04-
12',220,15);

```

```

--CLIENTE
INSERT INTO
CLIENTE(USERNAME,NOME,COGNOME,ETA,EMAIL,CITTA,VIA,CAP,NUMERO_T

```

```

ELEFONO) VALUES
('3TOGO', 'CARLO', 'DESICA', 20, 'decarl88@yahoo.it', 'NAPOLI', 'VIA_
_GARIBALDI', 80100, 3458945654);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('VSTORM26', 'VINCENZO', 'ESPOSITO', 31, 'vinsEsp26@yahoo.it', 'TOR
INO', 'VIA_CARDUCCI', 10094, 334907654);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('GHOST78', 'SIMONE', 'ANGELIS', 34, 'simo078@yahoo.it', 'NAPOLI', '
VIA_DANTE', 80100, 3458124654);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('VLACO23', 'CARLO', 'ROMOLO', 37, 'biancoCR2@gmail.com', 'ROMA', 'V
IA_SAN_GIOVANNI', 00042, 3348907654);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('DRAGO90', 'VINCENZO', 'ALTAMURA', 39, 'alvins3@libero.it', 'MILAN
O', 'VIA_ROMOLO', 20019, 3458907755);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('RERO', 'ALESSANDRO', 'RISSO', 24, 'aleRisso9@gmail.com', 'ROMA', '
VIA_ROMA', 00042, 3454907098);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('SCARPE01', 'ROBERTO', 'RUSSIO', 29, 'robRu8@libero.it', 'NAPOLI',
'VIA_REMO', 80100, 3228903654);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('CIRO45', 'CIRO', 'SCOTTO', 45, 'cirSC9@yahoo.it', 'TRIESTE', 'VIA_
CESARE', 34121, 3223907653);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('89YOULO', 'ALESSIA', 'TERRA', 20, 'aleRRa@gmail.com', 'ROMA', 'VIA_
LEOPARDI', 00042, 3994907654);
INSERT INTO
CLIENTE(USERNAME, NOME, COGNOME, ETA, EMAIL, CITTA, VIA, CAP, NUMERO_T
ELEFONO) VALUES
('JORDANUSER', 'ANTONIO', 'FABBRI', 22, 'antFab@yahoo.it', 'NAPOLI'
, 'VIA_MONTE', 80100, 3858907654);

```

```

INSERT INTO
CLIENTE(USERNAME,NOME,COGNOME,ETA,EMAIL,CITTA,VIA,CAP,NUMERO_T
ELEFONO) VALUES
('WALE2','MARIA','FARINA',23,'marFa3@yahoo.it','TRIESTE','VIA_
ROSARIO',34121,3788907454);
INSERT INTO
CLIENTE(USERNAME,NOME,COGNOME,ETA,EMAIL,CITTA,VIA,CAP,NUMERO_T
ELEFONO) VALUES
('EXTREME','FILIPPO','ELLIS',41,'Filel@yahoo.it','ROMA','VIA_S
TORIA',00042,3443407654);
INSERT INTO
CLIENTE(USERNAME,NOME,COGNOME,ETA,EMAIL,CITTA,VIA,CAP,NUMERO_T
ELEFONO) VALUES
('LIGHT009','ALESSANDRA','PAVON',19,'PAv0n@gmail.com','ROMA','
VIA_ONESTI',00042,3452233659);
INSERT INTO
CLIENTE(USERNAME,NOME,COGNOME,ETA,EMAIL,CITTA,VIA,CAP,NUMERO_T
ELEFONO) VALUES
('PENGUIN87','ALBERTO','MARTINI',19,'AlMar9@yahoo.it','FIRENZE
','VIA_SAN_VITO',50100,3466707654);
INSERT INTO
CLIENTE(USERNAME,NOME,COGNOME,ETA,EMAIL,CITTA,VIA,CAP,NUMERO_T
ELEFONO) VALUES
('TOTBLACK','GAIA','BIANCHI',19,'GABi@yahoo.it','MILANO','VIA_
VALORE',20019,3458781054);

```

```

--EFFETTUA
--OVVIAMENTE OGNI CLIENTE PUO' EFFETTUARE PIU' ABBONAMENTI
OLTRE A QUELLO FREE CHE E' GRATUITO,DI SEGUITO
--VENGONO ELENcate SOLO ALCUNI ESEMPI PER NON RENDERE IL
POPOLAMENTO TROPPO LUNGO
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','3TOGO',DATE'2020-05-24',DATE'2023-05-24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','VSTORM26',DATE'2020-05-24',DATE'2023-04-
24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','GHOST78',DATE'2020-03-24',DATE'2023-03-
24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A

```

```

BB) VALUES('FREE','VLAC023',DATE'2020-02-24',DATE'2023-02-
24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','DRAGO90',DATE'2020-01-24',DATE'2023-01-
24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','RERO',DATE'2020-06-24',DATE'2023-06-24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','SCARPE01',DATE'2020-03-21',DATE'2023-03-
21');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','89YOOLO',DATE'2020-06-24',DATE'2023-06-
24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','JORDANUSER',DATE'2020-03-23',DATE'2023-03-
23');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','WALE2',DATE'2020-02-24',DATE'2023-02-24');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','EXTREME',DATE'2020-01-23',DATE'2023-01-
23');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','LIGHT009',DATE'2020-01-23',DATE'2023-01-
23');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','PENGUIN87',DATE'2020-05-23',DATE'2023-05-
23');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','TOTBLACK',DATE'2020-05-25',DATE'2023-05-
25');
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('FREE','CIRO45',DATE'2020-05-26',DATE'2023-05-26');

INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('WOOD','GHOST78',DATE'2022-08-03',DATE'2022-09-
03');

```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('GLASS','GHOST78',DATE'2022-09-03',DATE'2022-12-
03');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('BRICK','89YOOL0',DATE'2021-10-19',DATE'2021-12-
19');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('PLASTIC','DRAGO90',DATE'2020-09-12',DATE'2021-02-
12');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('IRON','EXTREME',DATE'2022-05-10',DATE'2022-12-
10');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('MERCURY','JORDANUSER',DATE'2021-05-04',DATE'2022-
01-04');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('STEEL','RERO',DATE'2021-12-13',DATE'2022-10-13');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('SILVER','WALE2',DATE'2020-09-12',DATE'2021-08-
12');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('DIAMOND','VSTORM26',DATE'2022-05-23',DATE'2023-08-
23');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('PLATINUM','3TOGO',DATE'2021-12-23',DATE'2023-12-
23');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('PLATINUM','PENGUIN87',DATE'2021-11-23',DATE'2023-
11-23');
```

```
INSERT INTO
EFFETTUA(NOME_ABBONAMENTO,USERNAME,DATA_INIZIO_ABB,DATA_FINE_A
BB) VALUES('PLATINUM','TOTBLACK',DATE'2021-10-23',DATE'2023-
10-23');
```

```
--CARTA
--OVVIAMENTE OGNI CLIENTE DEVE INSERIRE UNA CARTA, DI SEGUITO
VENGONO INSERITE PERO' SOLO 15 CARTE
--AFFINCHE' OGNUNA SI RIFERISCA AD UNO SPECIFICO ACQUISTO E
PER NON RENDERE IL POPOLAMENTO TROPPO LUNGO
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(20739367,'3TOGO','ACQ01','VISA',DATE'2024-07-03',143);
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(34539398,'3TOGO','ACQ02','MASTERCARD',DATE'2025-08-03',893);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(55649367,'VSTORM26','ACQ03','VISA',DATE'2024-08-21',743);
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(09535367,'VSTORM26','ACQ04','PAYPAL',DATE'2023-09-21',323);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(67890564,'GHOST78','ACQ05','MASTERCARD',DATE'2024-07-
03',143);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(64538562,'VLACO23','ACQ06','PAYPAL',DATE'2028-11-20',039);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(14379167,'89YOOL0','ACQ07','PAYPAL',DATE'2024-08-03',198);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(59125500, 'JORDANUSER', 'ACQ08', 'PAYPAL', DATE '2026-07-03', 423);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(76091746, 'JORDANUSER', 'ACQ09', 'VISA', DATE '2024-07-03', 324);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(90511972, 'WALE2', 'ACQ10', 'VISA', DATE '2025-04-20', 432);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(50010695, 'EXTREME', 'ACQ11', 'MASTERCARD', DATE '2024-07-
03', 190);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(92313757, 'CIR045', 'ACQ12', 'PAYPAL', DATE '2025-07-23', 345);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(50010678, 'LIGHT009', 'ACQ13', 'MASTERCARD', DATE '2024-07-
02', 191);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(19894178, 'PENGUIN87', 'ACQ14', 'VISA', DATE '2028-12-15', 902);
```

```
INSERT INTO
CARTA(NUMERO_CARTA,USERNAME,CODICE_ACQUISTO,TIPO_CARTA,SCADENZ
A,CODICE_SICUREZZA) VALUES
(92313787, 'TOTBLACK', 'ACQ15', 'PAYPAL', DATE '2025-07-22', 346);
```

--CATALOGO

--OVVIAMENTE VENGONO ASSOCIATI CATALOGHI CHE DIFFERISCONO PER
GENERE E FASCIA D'ETA AD OGNI ABBONAMENTO,


```

--DI SEGUITO SE NE ELENCAO ALCUNI PER SMALTIRE MOLTE RIGHE IL
POPOLAMENTO
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO1MR','FREE','M','RAGAZZI');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO1MA','FREE','M','ADULTO');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO1FR','FREE','F','RAGAZZI');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO1FA','FREE','F','ADULTO');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO1UR','FREE','U','RAGAZZI');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO1UA','FREE','U','ADULTO');

INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO14MR','DIAMOND','M','RAGAZZI');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO14MA','DIAMOND','M','ADULTO');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO14UA','DIAMOND','U','ADULTO');

INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO15MR','PLATINUM','M','RAGAZZI');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO15MA','PLATINUM','M','ADULTO');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO15FR','PLATINUM','F','RAGAZZI');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO15FA','PLATINUM','F','ADULTO');
INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO15UR','PLATINUM','U','RAGAZZI');

```

```

INSERT INTO
CATALOGO(NOME_CATALOGO,NOME_ABBONAMENTO,GENERE,FASCIA_D_ETA)
VALUES ('CATALOGO15UA','PLATINUM','U','ADULTO');

--PRODOTTO
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD01','CATALOGO1UA','ACQ02','JORDAN',DATE'2021-12-04');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD02','CATALOGO1UA','ACQ03','NIKE',DATE'2021-12-05');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD03','CATALOGO15UA','ACQ01','DIOR',DATE'2021-12-06');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD04','CATALOGO1UA','ACQ04','YEEZY',DATE'2021-12-07');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD05','CATALOGO1UA','ACQ05','CACTUS_JACK',DATE'2021-12-
01');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD06','CATALOGO15UA','ACQ06','JORDAN',DATE'2021-12-02');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD07','CATALOGO1UR','ACQ07','JORDAN',DATE'2021-12-03');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD08','CATALOGO15UA','ACQ08','OFF_WHITE',DATE'2021-12-08');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD09','CATALOGO15UA','ACQ09','JORDAN',DATE'2021-12-09');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_

```

```

PUBBLICAZIONE) VALUES
('PRD10','CATALOGO1UA','ACQ10','ADIDAS',DATE'2021-12-10');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD11','CATALOGO1UA','ACQ09','MCQUEEN',DATE'2021-12-11');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD12','CATALOGO1UA','ACQ10','YEEZY',DATE'2021-12-12');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD13','CATALOGO1UA','ACQ11','YEEZY',DATE'2021-12-13');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD14','CATALOGO1UA','ACQ12','ADIDAS',DATE'2021-12-14');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD15','CATALOGO1UA','ACQ13','JORDAN',DATE'2021-12-15');

INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD16M','CATALOGO1MA','ACQ15','OVO',DATE'2021-12-16');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD16L','CATALOGO1MA','ACQ14','OVO',DATE'2021-12-16');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD16XL','CATALOGO1MA','ACQ02','OVO',DATE'2021-12-16');

INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD17S','CATALOGO1MR','ACQ01','THE_NORTH_FACE',DATE'2021-12-
20');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_
PUBBLICAZIONE) VALUES
('PRD17M','CATALOGO1MA','ACQ02','THE_NORTH_FACE',DATE'2021-12-
20');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_

```

```

PUBBLICAZIONE) VALUES
('PRD17L','CATALOGO1MA','ACQ03','THE_NORTH_FACE',DATE'2021-12-20');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD17XL','CATALOGO1MA','ACQ04','THE_NORTH_FACE',DATE'2021-12-20');

INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD18S','CATALOGO1MR','ACQ05','VLONE',DATE'2021-12-24');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD18M','CATALOGO1MA','ACQ06','VLONE',DATE'2021-12-24');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD18L','CATALOGO1MA','ACQ07','VLONE',DATE'2021-12-24');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD18XL','CATALOGO1MA','ACQ08','VLONE',DATE'2021-12-24');

INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD19S','CATALOGO1FR','ACQ01','SUPREME',DATE'2021-12-17');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD19M','CATALOGO1FA','ACQ01','SUPREME',DATE'2021-12-17');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD19L','CATALOGO1FA','ACQ03','SUPREME',DATE'2021-12-17');
INSERT INTO
PRODOTTO(ID_PRODOTTO,NOME_CATALOGO,CODICE_ACQUISTO,BRAND,DATA_PUBBLICAZIONE) VALUES
('PRD19XL','CATALOGO1FA','ACQ11','SUPREME',DATE'2021-12-17');

```

--CAPI HYPE

```

INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD16M','OVO_JACKET','M',220,'FELPA','COTONE');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD16L','OVO_JACKET','L',220,'FELPA','COTONE');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD16XL','OVO_JACKET','XL',220,'FELPA','COTONE');

```

```

INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD17S','NORTH_FACE_JACKET','S',70,'GIUBBINO','COTONE');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD17M','NORTH_FACE_JACKET','M',70,'GIUBBINO','COTONE');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD17L','NORTH_FACE_JACKET','L',70,'GIUBBINO','COTONE');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD17XL','NORTH_FACE_JACKET','XL',70,'GIUBBINO','COTONE');

```

```

INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD18S','VLONE_X_JUICE','S',180,'TSHIRT','CUOIO');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD18M','VLONE_X_JUICE','M',180,'TSHIRT','CUOIO');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD18L','VLONE_X_JUICE','L',180,'TSHIRT','CUOIO');

```

```

INSERT INTO

```

```

CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD18XL','VLONE_X_JUICE','XL',180,'TSHIRT','CUOIO');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD19S','SUPREME_FUTURA','S',150,'TSHIRT','STOFFA');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD19M','SUPREME_FUTURA','M',150,'TSHIRT','STOFFA');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD19L','SUPREME_FUTURA','L',150,'TSHIRT','STOFFA');
INSERT INTO
CAPI_HYPE(ID_PRODOTTO,NOME_CAPO,TAGLIA,PREZZO_CAPO,TIPOLOGIA,T
ESSUTO) VALUES
('PRD19XL','SUPREME_FUTURA','XL',150,'TSHIRT','STOFFA');

```

--SCARPE

```

INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD01','JORDAN_4_RETRO_MB','SNEACKERS','PELLE',430);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD02','DUNK_PANDA_HIGH','SNEACKERS','TESSUTO',180);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES ('PRD03','DIOR_B22','SNEACKERS','PELLE',900);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES ('PRD04','YEEZY_SLIDE_PURE','SLIPPERS','CUOIO',150);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD05','TRAVIS_SCOTT_CJ_BAROQUE','SNEACKERS','PELLE',500);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD06','JORDAN1_CHICAGO_HIGH','SNEACKERS','CUOIO',450);

```

```

INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD07','JORDAN1_UNIVERSITY_HIGH','SNEACKERS','TESSUTO',450);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD08','OFF_WHITE_X_JORDAN1_RETRO','SNEACKERS','TESSUTO',600
);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD09','JORDAN_6_RETRO_RED_OREO','SNEACKERS','CUOIO',270);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES ('PRD10','ADIDAS_OZELIA','SNEACKERS','TESSUTO',100);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD11','ALEXANDER_MCQUEEN','SNEACKERS','PELLE',450);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES ('PRD12','YEEZY_FOAM_RNNR','SLIPPERS','CUOIO',200);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES ('PRD13','YEEZY_BOOST_350','SNEACKERS','TELETE',370);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES ('PRD14','ADIDAS_OZWEEGO','SNEACKERS','TESSUTO',130);
INSERT INTO
SCARPE(ID_PRODOTTO,NOME_SCARPE,MODELLO,MATERIALE,PREZZO_SCARPE
) VALUES
('PRD15','JORDAN_1_RETRO_HIGH_OG','SNEACKERS','CUOIO',240);

```

--TAGLIA SCARPE

```

INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(36,10);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(37,15);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(38,20);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(39,25);

```

```
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(40,30);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(41,35);
```

```
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(42,40);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(43,45);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(44,50);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(45,55);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(46,60);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(47,65);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(48,70);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(49,75);
INSERT INTO TAGLIA_SCARPE(NUMERO_TAGLIA,PREZZO_TAGLIA) VALUES
(50,80);
```

```
--SUDDIVISE IN
--PRESI SOLO 2 PRODOTTI COME ESEMPIO
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',36);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',37);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',38);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',39);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',40);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',41);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',42);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',43);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',44);
```



```

INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',45);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',46);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',47);

```

```

INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',48);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',49);

```

```

INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD01',50);

```

```

INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',36);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',37);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',38);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',39);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',40);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',41);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',42);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',43);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',44);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',45);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',46);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',47);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',48);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',49);
INSERT INTO SUDDIVISE_IN(ID_PRODOTTO,NUMERO_TAGLIA)
VALUES('PRD02',50);

```

```

--FORNITURA
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN01','JORDAN','PRD01',DATE'2022-05-28',70,'40',8000);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN02','JORDAN','PRD01',DATE'2022-05-28',70,'41',8000);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN03','JORDAN','PRD01',DATE'2022-05-01',70,'42',8000);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN04','JORDAN','PRD01',DATE'2022-06-01',70,'42',8600);

INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN05','NIKE','PRD02',DATE'2022-03-28',70,'40',8000);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN06','NIKE','PRD02',DATE'2022-02-28',70,'40',8800);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN07','NIKE','PRD02',DATE'2022-05-07',70,'42',8400);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN08','NIKE','PRD02',DATE'2022-05-07',70,'43',8400);

INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN09','DIOR','PRD03',DATE'2022-04-26',70,'41',12000);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN10','DIOR','PRD03',DATE'2022-05-26',70,'41',12000);

INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN11','YEEZY','PRD04',DATE'2022-05-22',70,'46',8400);

```

```

INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN12','YEEZY','PRD04',DATE'2022-05-22',70,'44',8400);

```

```

INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN13','CACTUS_JACK','PRD05',DATE'2022-05-21',70,'45',9400);

```

```

INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN14','OVO','PRD16L',DATE'2022-05-08',70,'L',8800);
INSERT INTO
FORNITURA(ID_FORNITURA,AZIENDA_FORNITORE,ID_PRODOTTO,DATA_FORNITURA,QUANTITA,TAGLIA_FORNITA,PREZZO_FORNITURA) VALUES
('FRN15','OVO','PRD16M',DATE'2022-05-08',70,'M',8800);

```

```

--E' CONTENUTA IN
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ01','FRN01',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ01','FRN02',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ01','FRN03',70);

```

```

INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ02','FRN04',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ02','FRN05',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ02','FRN06',70);

```

```

INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ03','FRN07',70);

```

```

INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ03','FRN08',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ03','FRN09',70);

INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ04','FRN10',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ04','FRN11',70);

INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ05','FRN12',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ05','FRN13',70);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ05','FRN14',40);

INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ06','FRN14',30);
INSERT INTO
E CONTENUTA_IN(NUMERO_MAGAZZINO,ID_FORNITURA,QUANTITA_DISTRIB)
VALUES ('MGZ06','FRN15',70);

```

Creazione degli utenti

Di seguito verrà riportata la lista degli utenti presenti nel database:

```
--CREATE USER ADMIN  
CREATE USER ADMIN IDENTIFIED BY psw_admin ;  
GRANT ALL PRIVILEGES TO ADMIN ;
```

```
--CREATE USER CLIENTE  
CREATE USER CLIENTE IDENTIFIED BY psw_cliente ;  
GRANT CREATE SESSION TO CLIENTE ;
```

```
CREATE ROLE CLIENTI ;  
GRANT SELECT CLIENTE TO CLIENTI ;  
GRANT SELECT CARTA TO CLIENTI ;  
GRANT SELECT ABBONAMENTO TO CLIENTI ;  
GRANT SELECT RELEASE_DAY TO CLIENTI ;  
GRANT SELECT CATALOGO TO CLIENTI ;  
GRANT SELECT ACQUISTO TO CLIENTI ;  
GRANT SELECT PRODOTTO TO CLIENTI;  
GRANT CLIENTI TO CLIENTE ;
```

```
--CREATE USER GESTORE_VENDITE  
CREATE USER GESTORE_VENDITE IDENTIFIED BY psw_vendite ;  
GRANT CREATE SESSION TO GESTORE_VENDITE ;
```

```
CREATE ROLE VENDITE ;  
GRANT ALL CARTA TO VENDITE ;  
GRANT ALL ACQUISTO TO VENDITE ;  
GRANT ALL PROMO TO VENDITE ;
```

```
GRANT SELECT CLIENTE TO VENDITE;
```

```
--CREATE USER MANAGER_SERVIZI  
CREATE USER MANAGER_SERVIZI IDENTIFIED BY psw_servizi ;  
GRANT CREATE SESSION TO MANAGER_SERVIZI ;
```

```
CREATE ROLE SERVIZI ;
```

```
GRANT ALL RELEASE_DAY TO SERVIZI ;
GRANT ALL DA_ACCESSO TO SERVIZI ;
GRANT ALL EFFETTUA TO SERVIZI ;
GRANT ALL CATALOGO TO SERVIZI ;
```

```
GRANT SELECT ABBONAMENTO TO SERVIZI ;
GRANT SELECT CLIENTE TO SERVIZI ;
GRANT SELECT PRODOTTO TO SERVIZI ;
```

```
--CREATE USER MANAGER_LOGISTICA
CREATE USER MANAGER_LOGISTICA IDENTIFIED BY psw_logistica ;
GRANT CREATE SESSION TO MANAGER_LOGISTICA ;
```

```
CREATE ROLE LOGISTICA ;
GRANT ALL ON FORNITORE TO LOGISTICA ;
GRANT ALL ON FORNITURA TO LOGISTICA ;
GRANT ALL ON MAGAZZINO TO LOGISTICA ;
GRANT ALL ON E CONTENUTA_IN TO LOGISTICA ;
GRANT LOGISTICA TO MANAGER_LOGISTICA ;
```

```
--CREATE USER GESTORE_PRODOTTI
CREATE USER GESTORE_PRODOTTI IDENTIFIED BY psw_prodotti ;
GRANT CREATE SESSION TO GESTORE_PRODOTTI ;
```

```
CREATE ROLE PRODOTTI ;
GRANT ALL PRODOTTO TO PRODOTTI ;
GRANT ALL CAPI_HYPE TO PRODOTTI ;
GRANT ALL SCARPE TO PRODOTTI ;
GRANT ALL SUDDIVISE_IN TO PRODOTTI ;
GRANT ALL TAGLIA_SCARPE TO PRODOTTI ;
```

```
GRANT SELECT CATALOGO TO PRODOTTI ;
```

Triggers

Di seguito verranno elencati i triggers della nostra base di dati:

Trigger 1: UN_SOLO_ABB

```
--Trigger che confronta due abbonamenti
--Nel caso in cui i due abbonamenti, effettuati dallo stesso
cliente, hanno stesso nome
--e stesse date d'inizio e di fine, allora lancia un errore
per cui non si potrà inserire ulteriormente
--quell'abbonamento in quelle date

CREATE OR REPLACE TRIGGER UN_SOLO_ABB
BEFORE INSERT ON EFFETTUA
FOR EACH ROW
DECLARE
INVALID_SUB EXCEPTION;
SUB_NAME EFFETTUA.NOME_ABBONAMENTO%TYPE;
USERNM EFFETTUA.USERNAME%TYPE;
SUB BOOLEAN := FALSE; --FLAG

BEGIN
SELECT NOME_ABBONAMENTO, USERNAME INTO SUB_NAME, USERNM
FROM EFFETTUA
WHERE NOME_ABBONAMENTO = :NEW.NOME_ABBONAMENTO AND
DATA_INIZIO_ABB = :NEW.DATA_INIZIO_ABB AND
DATA_FINE_ABB = :NEW.DATA_FINE_ABB AND
USERNAME = :NEW.USERNAME;

--CONFRONTA DUE ABBONAMENTI, SE QUELLO INSERITO E' DIVERSO DA
QUELLO PRECEDENTEMENTE EFFETTUATO
--LA FLAG SARA' TRUE, ALTRIMENTI LANCIA L'ERRORE
IF SUB_NAME != :NEW.NOME_ABBONAMENTO AND USERNM =
:NEW.USERNAME THEN
SUB:=TRUE;

ELSE
RAISE INVALID_SUB;
END IF;
```

```

--SE IL NUOVO ABBONAMENTO INSERITO HA DATE DIFFERENTI DA
QUELLO GIA' INSERITO IN PRECEDENZA,
--LA FLAG SARA' TRUE,ALTRIMENTI LANCIATA L'ERRORE
IF SYSDATE != :NEW.DATA_INIZIO_ABB AND SYSDATE !=
:NEW.DATA_FINE_ABB THEN
SUB:=TRUE;
ELSE
RAISE INVALID_SUB;
END IF;

--GESTIONE DELLE VARIE ECCEZIONI
EXCEPTION
WHEN INVALID_SUB THEN
RAISE_APPLICATION_ERROR(-20008, 'ABBONAMENTO CON DATE E NOME
IDENTICI AL PRECEDENTE');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('ABBONAMENTI CONFRONTATI,NESSUN
ABBONAMENTO TROVATO');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008, 'ABBONAMENTO
CONFRONTATO,RISCONTRI CON DATI NON DEL CLIENTE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008, 'PROBLEMA RISCONTRATO');
END;

```

Trigger 2: CARTA_SCADUTA

```

--Trigger che controlla se una carta per effettuare acquisti
sia scaduta o no,
--nel caso in cui la carta sia scaduta lancia un errore

CREATE OR REPLACE TRIGGER CARTA_SCADUTA
BEFORE INSERT ON CARTA
FOR EACH ROW
DECLARE
INVALID_CARD EXCEPTION;
SCADUTA CARTA.SCADENZA%TYPE;

BEGIN
--SE LA DATA DI SISTEMA E' MAGGIORE ALLA DATA DELLA SCADENZA
DELLA CARTA,ALLORA
--E' IMPOSSIBILE UTILIZZARE QUELLA CARTA E VERRA' LANCIATO UN
ERRORE

```



```

IF :NEW.SCADENZA < SYSDATE THEN
RAISE INVALID_CARD;
END IF;

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_CARD THEN
RAISE_APPLICATION_ERROR(-20008, 'CARTA SCADUTA, IMPOSSIBILE
INSERIRLA');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('SCADENZA CONTROLLATA, CARTA NON
SCADUTA');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008, 'PIU SCADENZE CONFRONTATE, CARTA
NON SCADUTA');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008, 'CARTA SCADUTA, IMPOSSIBILE
INSERIRLA');
END;

```

Trigger 3: RESO

```

--Trigger che controlla se un cliente può eseguire il
reso. Esso è possibile se
--la data in cui lo richiede è minore a 31 giorni da quando ha
effettuato l'acquisto.
--Nel caso contrario verrà lanciato un errore

```

```

CREATE OR REPLACE TRIGGER RESO
BEFORE INSERT ON ACQUISTO
FOR EACH ROW
DECLARE
INVALID_RESO EXCEPTION;
DATA_RESO ACQUISTO.DATA_ACQUISTO%TYPE;
COD_ACQUISTO ACQUISTO.CODICE_ACQUISTO%TYPE;
RES BOOLEAN := FALSE; --FLAG

BEGIN
SELECT DATA_ACQUISTO, CODICE_ACQUISTO INTO DATA_RESO,
COD_ACQUISTO
FROM ACQUISTO
WHERE DATA_ACQUISTO = :NEW.DATA_ACQUISTO AND
CODICE_ACQUISTO = :NEW.CODICE_ACQUISTO;

```

```

--SE TRA LA DATA D'ACQUISTO E LA DATA DI SISTEMA SONO PASSATI
MENO DI 31 GIORNI,
--ALLORA E' POSSIBILE ESEGUIRE IL RESO, ALTRIMENTI VERRA'
LANCIATO UN ERRORE
IF (SYSDATE - :NEW.DATA_ACQUISTO) < 31 AND
COD_ACQUISTO = :NEW.CODICE_ACQUISTO THEN
RES:=TRUE;
ELSE
RAISE INVALID_RESO;
END IF;

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_RESO THEN
RAISE_APPLICATION_ERROR(-20008,'31 GIORNI O PIU,IMPOSSIBILE
ESEGUIRE IL RESO');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NUOVA DATA');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'ULTERIORE DATA TROVATA');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;

```

Trigger 4: UN_SOLO_CATALOGO

```

--Trigger che controlla se un prodotto è già contenuto in un
catalogo,
--e se ciò risultasse vero esso non può essere aggiunto
ulteriormente
--all'interno di un altro catalogo

CREATE OR REPLACE TRIGGER UN_SOLO_CATALOGO
BEFORE INSERT ON PRODOTTO
FOR EACH ROW
DECLARE
INVALID_CATALOGO EXCEPTION;
CATALOGUE PRODOTTO.NOME_CATALOGO%TYPE;
PROD_CATALOG PRODOTTO.ID_PRODOTTO%TYPE;

BEGIN
SELECT NOME_CATALOGO, ID_PRODOTTO INTO CATALOGUE, PROD_CATALOG
FROM PRODOTTO

```

```
WHERE NOME_CATALOGO = :NEW.NOME_CATALOGO AND  
ID_PRODOTTO = :NEW.ID_PRODOTTO;
```

```
--SE IL NOME DI UN PRODOTTO CORRISPONDE AD UN ALTRO PRODOTTO,  
CONTENUTO IN UN ALTRO CATALOGO,  
--ALLORA SARA' IMPOSSIBILE INSERIRE QUEL PRODOTTO IN UN ALTRO  
CATALOGO  
IF :NEW.NOME_CATALOGO = CATALOGUE AND :NEW.ID_PRODOTTO =  
PROD_CATALOG THEN  
RAISE INVALID_CATALOGO;  
END IF;  
  
--GESTIONE DELLE ECCEZIONI  
EXCEPTION  
WHEN INVALID_CATALOGO THEN  
RAISE_APPLICATION_ERROR(-20008, 'PRODOTTO GIA CONTENUTO IN UN  
CATALOGO');  
WHEN NO_DATA_FOUND THEN  
DBMS_OUTPUT.PUT_LINE('PRODOTTO NON CONTENUTO IN NESSUN  
CATALOGO');  
WHEN TOO_MANY_ROWS THEN  
RAISE_APPLICATION_ERROR(-20008, 'ULTERIORE PRODOTTO TROVATO');  
WHEN OTHERS THEN  
RAISE_APPLICATION_ERROR(-20008, 'PROBLEMA RISCONTRATO');  
END;
```

Trigger 5: RELEASE_YEAR

--Trigger che gestisce i release day e che cerca di contenere
il numero di release day presentati in un anno.
--Nel caso in cui i release day inseriti in un anno(in questo
caso è stato preso come riferimento il 2022)
--siano maggiori a 20, il contatore si fermerà e lancerà un
errore

```
CREATE OR REPLACE TRIGGER RELEASE_YEAR  
BEFORE INSERT ON RELEASE_DAY  
FOR EACH ROW  
DECLARE  
INVALID_RELEASE EXCEPTION;  
CONTATORE NUMBER;  
ANNO_CORRENTE NUMBER;
```

```

BEGIN
SELECT EXTRACT (year FROM DATA_RELEASE) AS year,
      COUNT(DATA_RELEASE)
      INTO ANNO_CORRENTE, CONTATORE
FROM RELEASE_DAY
GROUP BY EXTRACT(year FROM DATA_RELEASE);

--SE DURANTE IL CORSO DELL'ANNO VERRANNO CONTATI PIU' DI 20
RELEASE DAY
--VERRA' LANCIATO UN ERRORE
IF CONTATORE > 20
THEN
RAISE INVALID_RELEASE;
END IF;

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_RELEASE THEN
RAISE_APPLICATION_ERROR(-20008,'TROPPE RELEASE PROGRAMMATE IN
UN ANNO');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('RELEASE NON TROVATA');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'ULTERIORE RELEASE TROVATA');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;

```

Trigger 6: CAPACITA_MAX

--Trigger che gestisce il numero di scorte all'interno di un magazzino. Se il numero totale delle scorte --distribuite all'interno di quel magazzino sarà maggiore della capienza del magazzino stesso --allora verrà lanciato un errore, che segnala che è stata oltrepassata la capienza

```

CREATE OR REPLACE TRIGGER CAPACITA_MAX
BEFORE INSERT ON E CONTENUTA_IN
FOR EACH ROW
DECLARE
INVALID_CAPACITA EXCEPTION;
QUANTITA E CONTENUTA_IN.QUANTITA_DISTRIB%TYPE;
CAPIENZA MAGAZZINO.CAPIENZA%TYPE;

```

```

BEGIN
SELECT
SUM(E CONTENUTA_IN.QUANTITA_DISTRIB),MAX(MAGAZZINO.CAPIENZA)
INTO QUANTITA,CAPIENZA
FROM E CONTENUTA_IN JOIN MAGAZZINO ON
E CONTENUTA_IN.NUMERO_MAGAZZINO = MAGAZZINO.NUMERO_MAGAZZINO

GROUP BY E CONTENUTA_IN.NUMERO_MAGAZZINO
HAVING E CONTENUTA_IN.NUMERO_MAGAZZINO =
:NEW.NUMERO_MAGAZZINO;

--SE LA SOMMA TRA LA QUANTITA' PRESENTE IN MAGAZZINO E LA
QUANTITA' DA INSERIRE
--SUPERA LA CAPIENZA DEL MAGAZZINO
--ALLORA SARA' IMPOSSIBILE INSERIRE ALTRE SCORTE
IF (QUANTITA + :NEW.QUANTITA_DISTRIB) > CAPIENZA
THEN
RAISE INVALID_CAPACITA;
END IF;

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_CAPACITA THEN
RAISE_APPLICATION_ERROR(-20008,'SCORTE SUPERIORI ALLA
CAPIENZA');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('SPAZIO DISPONIBILE');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'PIU SCORTE CONTENUTE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;

```

Scheduler

Di seguito verrà inserito lo scheduler:

```
BEGIN
--CREO UN JOB CHE VERRA' ESEGUITO OGNI AD OGNI PRIMO GIORNO
DEL MESE
--TALE JOB EFFETTUERA' UN'OPERAZIONE DI PULIZIA DEL DATABASE
ELIMINANDO LE PROMO ORMAI SCADUTE
DBMS_SCHEDULER.create_job(

    --DEFINISCO NOME DEL JOB
    job_name => 'promo_scad_delete',
    --DEFINISCO TIPO DEL JOB
    job_type => 'PLSQL_BLOCK',
    --WHERE CI DA LA POSSIBILITA' DI ELIMINARE SOLO LE PROMO

SCADURE
    job_action => 'BEGIN

                                DELETE FROM PROMO
                                WHERE

DATA_FINE_PROMO<SYSDATE;

                                END;',

    --DATA INIZIO JOB
    start_date => TO_DATE('01-SET-2020','DD-MON-YYYY'),
    --FREQUENZA TEMPORALE DI ATTIVAZIONE
    repeat_interval => 'FREQ=MONTHLY',
    enabled => TRUE,
    comments => 'Cancellazione promo scadute');
END;
```

Stored Procedures

Di seguito verranno elencate le stored procedures della nostra base di dati:

Stored Procedure 1: AUMENTO_PREZZO_SCARPE

--Procedura che aggiorna di una percentuale il prezzo di un paio di scarpe, dopo un anno
--dal rilascio(data di pubblicazione) sul sito

```
CREATE OR REPLACE PROCEDURE AUMENTO_PREZZO_SCARPE
(DATA_PUB PRODOTTO.DATA_PUBBLICAZIONE%TYPE ,ID_PR
PRODOTTO.ID_PRODOTTO%TYPE,PERCENTUALE NUMBER DEFAULT 1)
IS
INVALID_PREZZO EXCEPTION;
NEW_PRICE_SCARPE SCARPE.PREZZO_SCARPE%TYPE;
DATA_SCARPE PRODOTTO.DATA_PUBBLICAZIONE%TYPE;

BEGIN
--AGGIORNAMENTO PREZZO SCARPE
SELECT SCARPE.PREZZO_SCARPE, PRODOTTO.DATA_PUBBLICAZIONE INTO
NEW_PRICE_SCARPE, DATA_SCARPE
FROM SCARPE
JOIN PRODOTTO ON SCARPE.ID_PRODOTTO = PRODOTTO.ID_PRODOTTO
WHERE PRODOTTO.DATA_PUBBLICAZIONE = DATA_PUB AND
PRODOTTO.ID_PRODOTTO = ID_PR;

IF((SYSDATE-DATA_PUB)>=365) THEN
UPDATE SCARPE
SET SCARPE.PREZZO_SCARPE = SCARPE.PREZZO_SCARPE * PERCENTUALE;
COMMIT;

DBMS_OUTPUT.PUT_LINE('PREZZO SCARPE AGGIORNATO');
END IF;

IF((SYSDATE-DATA_PUB)<365) THEN
DBMS_OUTPUT.PUT_LINE('PREZZO SCARPE NON AGGIORNATO POICHE NON
TRASCORSI 365 GIORNI');
ROLLBACK;
END IF;
```

```

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_PREZZO THEN
RAISE_APPLICATION_ERROR(-20008,'PREZZO ERRATO');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NESSUN PRODOTTO PUBBLICATO IN QUELLA
DATA');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'ULTERIORE PREZZO DA
AGGIUNGERE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;

```

Stored Procedure 2: AUMENTO_PREZZO_CAPI

```

--Procedura che aggiorna di una percentuale il prezzo di un
capo, dopo un anno
--dal rilascio(data di pubblicazione) sul sito

```

```

CREATE OR REPLACE PROCEDURE AUMENTO_PREZZO_CAPI
(DATA_PUB PRODOTTO.DATA_PUBBLICAZIONE%TYPE ,ID_PR
PRODOTTO.ID_PRODOTTO%TYPE,PERCENTUALE NUMBER DEFAULT 1)
IS
INVALID_PREZZO EXCEPTION;
NEW_PRICE_CAPI CAPI_HYPE.PREZZO_CAPO%TYPE;
DATA_CAPI PRODOTTO.DATA_PUBBLICAZIONE%TYPE;

BEGIN
--AGGIORNAMENTO PREZZO CAPI
SELECT CAPI_HYPE.PREZZO_CAPO, PRODOTTO.DATA_PUBBLICAZIONE INTO
NEW_PRICE_CAPI, DATA_CAPI
FROM CAPI_HYPE
JOIN PRODOTTO ON CAPI_HYPE.ID_PRODOTTO = PRODOTTO.ID_PRODOTTO
WHERE PRODOTTO.DATA_PUBBLICAZIONE = DATA_PUB AND
PRODOTTO.ID_PRODOTTO = ID_PR;

IF((SYSDATE-DATA_PUB)>=365) THEN
UPDATE CAPI_HYPE
SET CAPI_HYPE.PREZZO_CAPO = CAPI_HYPE.PREZZO_CAPO *
PERCENTUALE;
COMMIT;
DBMS_OUTPUT.PUT_LINE('PREZZO CAPI_HYPE AGGIORNATO');
END IF;

```



```

IF((SYSDATE-DATA_PUB)<365) THEN
DBMS_OUTPUT.PUT_LINE('PREZZO CAPI_HYPE NON AGGIORNATO POICHE
NON TRASCORSI 365 GIORNI');
ROLLBACK;
END IF;

EXCEPTION
WHEN INVALID_PREZZO THEN
RAISE_APPLICATION_ERROR(-20008,'PREZZO ERRATO');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NESSUN PRODOTTO PUBBLICATO IN QUELLA
DATA');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'ULTERIORE PREZZO DA
AGGIUNGERE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;

```

Stored Procedure 3: RINNOVO_FREE

--Dopo 36 mesi l'abbonamento free si rinnova automaticamente
(1095 GIORNI)

```

CREATE OR REPLACE PROCEDURE RINNOVO_FREE
(DATA_FINE EFFETTUA.DATA_FINE_ABB%TYPE, NOME_ABB
EFFETTUA.NOME_ABBONAMENTO%TYPE)
IS
NEW_DATA_FINE EFFETTUA.DATA_FINE_ABB%TYPE;
NEW_NOME_ABB EFFETTUA.NOME_ABBONAMENTO%TYPE;
INVALID_NOME EXCEPTION;

BEGIN
SELECT EFFETTUA.DATA_FINE_ABB, EFFETTUA.NOME_ABBONAMENTO INTO
NEW_DATA_FINE, NEW_NOME_ABB
FROM EFFETTUA
WHERE EFFETTUA.DATA_FINE_ABB = DATA_FINE AND
EFFETTUA.NOME_ABBONAMENTO = NOME_ABB;

--RINNOVO LA DATA FINE DELL'ABBONAMENTO FREE,INCREMENTANDO LA
DATA FINE DI 1095 GIORNI (3 ANNI)
IF(NOME_ABB = 'FREE' AND DATA_FINE <= SYSDATE) THEN

```

```

UPDATE EFFETTUA
SET EFFETTUA.DATA_FINE_ABB = EFFETTUA.DATA_FINE_ABB + 1095;
COMMIT;
DBMS_OUTPUT.PUT_LINE('RINNOVO AUTOMATICO DI ABBONAMENTO FREE
EFFETTUATO.');
```

```

END IF;

IF(DATA_FINE != SYSDATE)THEN
DBMS_OUTPUT.PUT_LINE('RINNOVO DI ABBONAMENTO FREE NON
EFFETTUATO POICHE GIA PRESENTE UNO VALIDO.');
```

```

END IF;
IF(NOME_ABB != 'FREE') THEN
ROLLBACK;
END IF;

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_NOME THEN
RAISE_APPLICATION_ERROR(-20008,'NOME ABBONAMENTO ERRATO');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NESSUN ABBONAMENTO TROVATO');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'ULTERIORE ABBONAMENTO DA
AGGIUNGERE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;
```

Stored Procedure 4: PREZZO_FINALE_SCARPE

--Calcola il prezzo finale di una scarpa sommando taglia e prezzo

```

CREATE OR REPLACE PROCEDURE PREZZO_FINALE_SCARPE
(PREZZO_SCARPE_BASE SCARPE.PREZZO_SCARPE%TYPE, PREZZO_T
TAGLIA_SCARPE.PREZZO_TAGLIA%TYPE)
IS
P_TAGLIA TAGLIA_SCARPE.PREZZO_TAGLIA%TYPE;
P_SCARPE_BASE SCARPE.PREZZO_SCARPE%TYPE;
INVALID_COST EXCEPTION;

BEGIN
SELECT SCARPE.PREZZO_SCARPE, TAGLIA_SCARPE.PREZZO_TAGLIA INTO
P_SCARPE_BASE, P_TAGLIA
```

```

FROM (SCARPE JOIN SUDDIVISE_IN ON SCARPE.ID_PRODOTTO =
SUDDIVISE_IN.ID_PRODOTTO)
JOIN TAGLIA_SCARPE ON TAGLIA_SCARPE.NUMERO_TAGLIA =
SUDDIVISE_IN.NUMERO_TAGLIA
WHERE TAGLIA_SCARPE.PREZZO_TAGLIA = PREZZO_T AND
SCARPE.PREZZO_SCARPE = PREZZO_SCARPE_BASE;

--AGGIORNIAMO IL PREZZO DELLE SCARPE AGGIUNGENDO IL PREZZO
DELLA TAGLIA
IF (PREZZO_SCARPE_BASE > 0 AND PREZZO_T > 0) THEN
UPDATE SCARPE
SET SCARPE.PREZZO_SCARPE = SCARPE.PREZZO_SCARPE + P_TAGLIA
--ENTRAMBI I PREZZI DEVONO OVVIAMENTE ESSERE MAGGIORI DI 0
--AFFINCHE' AVVENGA UNA VARIAZIONE DEL PREZZO
WHERE SCARPE.PREZZO_SCARPE = PREZZO_SCARPE_BASE;
COMMIT;
DBMS_OUTPUT.PUT_LINE('PREZZO SCARPA AGGIORNATO. ');
END IF;

IF (PREZZO_SCARPE_BASE <= 0 AND PREZZO_T <= 0) THEN
ROLLBACK;
END IF;

--GESTIONE DELLE ECCEZIONI
EXCEPTION
WHEN INVALID_COST THEN
RAISE_APPLICATION_ERROR(-20008, 'PREZZO INSERITO NON VALIDO ');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NUOVO PREZZO TROVATO');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008, 'TROPPE RIGHE IN INPUT');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008, 'PROBLEMA RISCONTRATO');
END;

```

Stored Procedure 5: GESTIONE_FORNITORE

```

--Procedura che elimina i dati di un fornitore, tranne quello
dell'azienda, se il fornitore
--non effettua consegne da 90 giorni o più

```

```

CREATE OR REPLACE PROCEDURE GESTIONE_FORNITORE

```

```

(NOME_F FORNITORE.NOME_FORNITORE%TYPE, COGNOME_F
FORNITORE.COGNOME_FORNITORE%TYPE, DT_ULTIMA_FORNT
FORNITURA.DATA_FORNITURA%TYPE)

IS
DATA_FOR FORNITURA.DATA_FORNITURA%TYPE;

NOME_FRN FORNITORE.NOME_FORNITORE%TYPE
COGNOME_FRN FORNITORE.COGNOME_FORNITORE%TYPE;
AZ_FORNITORE FORNITORE.AZIENDA_FORNITORE%TYPE;
--GESTIONE DEGLI ERRORI SIA PER FORNITURA CHE PER FORNITORE(2
EXCEPTION)
INVALID_FORNITORE EXCEPTION;
INVALID_FORNITURA EXCEPTION;

BEGIN
SELECT DISTINCT FORNITORE.NOME_FORNITORE,
FORNITORE.COGNOME_FORNITORE, FORNITURA.DATA_FORNITURA,
FORNITORE.AZIENDA_FORNITORE
INTO NOME_FRN, COGNOME_FRN, DATA_FOR, AZ_FORNITORE
FROM FORNITORE JOIN FORNITURA ON FORNITORE.AZIENDA_FORNITORE =
FORNITURA.AZIENDA_FORNITORE
WHERE FORNITORE.NOME_FORNITORE = NOME_F AND
FORNITORE.COGNOME_FORNITORE = COGNOME_F
AND FORNITURA.DATA_FORNITURA = DT_ULTIMA_FORNT;

--SE PASSANO PIU' DI 90 GIORNI DALL'ULTIMA CONSEGNA DEL
FORNITORE
--VENGONO CANCELLATI NOME,COGNOME E RECAPITO TELEFONICO DEL
FORNITORE
IF (SYSDATE - DT_ULTIMA_FORNT) > 90
THEN
UPDATE FORNITORE
--ELIMINIAMO NOME,COGNOME E RECAPITO TELEFONICO DEL FORNITORE
LASCIANDO L'AZIENDA
--ESSENZIALE PER LE TABELLE COLLEGATE A FORNITORE.
SET FORNITORE.NOME_FORNITORE = ' ',FORNITORE.COGNOME_FORNITORE
= ' ', FORNITORE.RECAPITO_TEL = 0
WHERE FORNITORE.NOME_FORNITORE = NOME_F AND
FORNITORE.COGNOME_FORNITORE = COGNOME_F;
COMMIT;
END IF;

IF((SYSDATE - DT_ULTIMA_FORNT) <= 90) THEN
ROLLBACK;
END IF;

```

```

--GESTIONE DELLE ECCEZIONI
EXCEPTION
--GESTIONE ECCEZIONE FORNITORE
WHEN INVALID_FORNITORE THEN
RAISE_APPLICATION_ERROR(-20008, 'FORNITORE INSERITO NON VALIDO
');
--GESTIONE ECCEZIONE FORNITURA

WHEN INVALID_FORNITURA THEN
RAISE_APPLICATION_ERROR(-20008, 'FORNITURA INSERITA NON VALIDA
');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('IL FORNITORE INSERITO NON HA INVIATO UNA
FORNITURA IN QUESTA DATA');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008, 'ULTERIORE DATO DA
AGGIUNGERE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008, 'PROBLEMA RISCONTRATO');
END;

```

Stored Procedure 6: ACQUISTO_CON_SCONTO

--Procedura che sottrae il valore dello sconto della promo dall'importo dell'acquisto

```

CREATE OR REPLACE PROCEDURE ACQUISTO_CON_SCONTO
(PREZZO_ACQUISTO ACQUISTO.IMPORTO_ACQUISTO%TYPE,
SCONTO_APPLICATO PROMO.SCONTO%TYPE)
IS
SCONTO_EFFETTIVO PROMO.SCONTO%TYPE;
ACQUISTO_EFFETTIVO ACQUISTO.IMPORTO_ACQUISTO%TYPE;
INVALID_ACQ EXCEPTION;

BEGIN
SELECT PROMO.SCONTO, ACQUISTO.IMPORTO_ACQUISTO INTO
SCONTO_EFFETTIVO, ACQUISTO_EFFETTIVO
FROM PROMO JOIN ACQUISTO ON PROMO.CODICE_PROMO =
ACQUISTO.CODICE_PROMO
WHERE PROMO.SCONTO = SCONTO_APPLICATO AND
ACQUISTO.IMPORTO_ACQUISTO = PREZZO_ACQUISTO;

IF (ACQUISTO_EFFETTIVO > 0 AND SCONTO_EFFETTIVO > 0) THEN
UPDATE ACQUISTO

```

```

SET ACQUISTO.IMPORTO_ACQUISTO = ACQUISTO.IMPORTO_ACQUISTO -
SCONTO_EFFETTIVO
--AFFINCHE' SIA COERENTE TALE PROCEDURA L'IMPORTO
DELL'ACQUISTO E LO SCONTO_EFFETTIVO DEVONO RISULTARE

--MAGGIORI DI 0
WHERE ACQUISTO.IMPORTO_ACQUISTO = PREZZO_ACQUISTO;
COMMIT;
DBMS_OUTPUT.PUT_LINE('VALORE SCONTO SOTTRATTO ALL IMPORTO
ACQUISTO');
END IF;

IF (ACQUISTO_EFFETTIVO <= 0 AND SCONTO_EFFETTIVO <= 0) THEN
ROLLBACK;
END IF;

EXCEPTION
--GESTIONE ERRORE ACQUISTO (se non inserito precedentemente)
WHEN INVALID_ACQ THEN
RAISE_APPLICATION_ERROR(-20008,'ACQUISTO INSERITO NON VALIDO
');
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('NESSUN ACQUISTO TROVATO');
WHEN TOO_MANY_ROWS THEN
RAISE_APPLICATION_ERROR(-20008,'ULTERIORE ACQUISTO DA
AGGIUNGERE');
WHEN OTHERS THEN
RAISE_APPLICATION_ERROR(-20008,'PROBLEMA RISCONTRATO');
END;

```

Query

Di seguito verranno riportate alcune Queries della nostra base di dati:

```
--VISUALIZZA NOME,COGNOME,TIPO_CARTA E DATA_ACQUISTO DI CHI HA  
EFFETTUATO UN ACQUISTO  
--NEGL'ULTIMI 3 MESI
```

```
SELECT CLIENTE.NOME , CLIENTE.COGNOME , CARTA.TIPO_CARTA ,  
ACQUISTO.DATA_ACQUISTO  
FROM ((CARTA  
INNER JOIN CLIENTE  
ON CARTA.USERNAME = CLIENTE.USERNAME)  
INNER JOIN ACQUISTO  
ON CARTA.CODICE_ACQUISTO = ACQUISTO.CODICE_ACQUISTO)  
WHERE (SYSDATE - ACQUISTO.DATA_ACQUISTO) < 90  
ORDER BY ACQUISTO.DATA_ACQUISTO;
```

```
-----  
--VISUALIZZA NOME,COGNOME,USERNAME DI CHI HA EFFETTUATO UN  
ACQUISTO(CODICE_ACQUISTO),  
--VISUALIZZANDO ANCHE LA CARTA UTILIZZATA, DI UNA  
SCARPA(NOME_SCARPA)  
--CON PREZZO SUPERIORE A 400  
--(VISUALIZZANDO DI CONSEGUENZA ANCHE PREZZO,NOME_SCARPE E  
CODICE_ACQUISTO)
```

```
SELECT DISTINCT CLI.NOME, CLI.COGNOME, CLI.USERNAME,  
SC.NOME_SCARPE, CAR.TIPO_CARTA, SC.PREZZO_SCARPE,  
ACQ.CODICE_ACQUISTO  
FROM (((PRODOTTO PROD JOIN SCARPE SC ON PROD.ID_PRODOTTO =  
SC.ID_PRODOTTO)  
JOIN ACQUISTO ACQ  
ON PROD.CODICE_ACQUISTO = ACQ.CODICE_ACQUISTO)  
JOIN CARTA CAR  
ON CAR.CODICE_ACQUISTO = ACQ.CODICE_ACQUISTO)  
JOIN CLIENTE CLI  
ON CLI.USERNAME = CAR.USERNAME  
WHERE SC.PREZZO_SCARPE > 399  
ORDER BY SC.PREZZO_SCARPE
```

--VISUALIZZA NUMERO_MAGAZZINO, LA CAPIENZA E IL TOTALE DEI
PRODOTTI CONTENUTI
--ALL'INTERNO DI QUEI MAGAZZINO

SELECT
MA.NUMERO_MAGAZZINO,MAX(MA.CAPIENZA),SUM(CONT.QUANTITA_DISTRIB
)
FROM E_CONTENUTA_IN CONT JOIN MAGAZZINO MA ON
CONT.NUMERO_MAGAZZINO = MA.NUMERO_MAGAZZINO
GROUP BY MA.NUMERO_MAGAZZINO
ORDER BY MA.NUMERO_MAGAZZINO

--VISUALIZZA USERNAME, NOME, COGNOME, DATA_INIZIO E DATA_FINE
DI TUTTI COLORO
--CHE HANNO ACQUISTATO UN ABBONAMENTO PLATINIUM L'ULTIMO ANNO

SELECT EFF.NOME_ABBONAMENTO,EFF.DATA_INIZIO_ABB,
EFF.DATA_FINE_ABB, CLI.USERNAME, CLI.NOME, CLI.COGNOME
FROM EFFETTUA EFF JOIN CLIENTE CLI ON EFF.USERNAME =
CLI.USERNAME
WHERE NOME_ABBONAMENTO = 'PLATINUM' AND
(SYSDATE - EFF.DATA_INIZIO_ABB) < 365
ORDER BY DATA_INIZIO_ABB

Viste

Di seguito verranno riportate alcune viste della nostra base di dati, e i corrispettivi utenti che le utilizzeranno:

User: CLIENTE

```
--VISTA CHE VISUALIZZA USERNAME,NOME,COGNOME ED ETA' DEI  
CLIENTI  
--CON L'ETA' PIU' ALTA
```

```
CREATE VIEW ETA_MAX_MIN AS  
SELECT USERNAME, NOME, COGNOME, ETA  
FROM CLIENTE  
WHERE ETA = (SELECT MAX(ETA) FROM CLIENTE)
```

User: CLIENTE

```
----VISTA CHE VISUALIZZA USERNAME,NOME,COGNOME ED ETA' DEI  
CLIENTI  
--CON L'ETA' PIU' BASSA
```

```
CREATE VIEW ETA_MAX_MIN AS  
SELECT USERNAME, NOME, COGNOME, ETA  
FROM CLIENTE  
WHERE ETA = (SELECT MIN(ETA) FROM CLIENTE)
```

User: GESTORE_VENDITE

```
--VISTA CHE VISUALIZZA CODICE PROMO, DATA INIZIO E DATA FINE  
DELLE PROMO  
--CHE INIZIANO DAL 1° MAGGIO 2022 IN POI
```

```
CREATE VIEW PROSSIME_PROMO AS  
SELECT CODICE_PROMO, DATA_INIZIO_PROMO, DATA_FINE_PROMO  
FROM PROMO  
WHERE DATA_INIZIO_PROMO >= DATE '2022-05-01'
```

User: GESTORE_VENDITE

```
--VISTA CHE VISUALIZZA USERNAME, CODICE ACQUISTO E DATA  
ACQUISTO
```

```
--DI TUTTI GLI ACQUISTI EFFETTUATI NEL 2022
```

```
CREATE VIEW UTENTI_ACQUISTI AS  
SELECT CARTA.CODICE_ACQUISTO, CARTA.USERNAME,  
ACQUISTO.DATA_ACQUISTO  
FROM CARTA JOIN ACQUISTO ON CARTA.CODICE_ACQUISTO =  
ACQUISTO.CODICE_ACQUISTO  
WHERE ACQUISTO.DATA_ACQUISTO >= DATE '2022-01-01'
```

User: GESTORE_PRODOTTI

```
--VISTA CHE VISUALIZZA NOME SCARPE, MODELLO E PREZZO DI TUTTE  
LE SNEACKERS PRESENTI
```

```
--NEL DATABASE
```

```
CREATE VIEW SNEACKERS AS  
SELECT NOME_SCARPE, MODELLO, PREZZO_SCARPE  
FROM SCARPE  
WHERE MODELLO = 'SNEACKERS'
```

User: GESTORE_PRODOTTI

```
--VISTA CHE VISUALIZZA NOME, TAGLIA E PREZZO DEI CAPI CON  
--PREZZO SUPERIORE AI 500€
```

```
CREATE VIEW CAPO_PREZZO AS  
SELECT NOME_CAPO, TAGLIA, PREZZO_CAPO  
FROM CAPI_HYPE  
WHERE PREZZO_CAPO >= 150
```

User: MANAGER_LOGISTICA

--VISTA CHE VISUALIZZA NOME, COGNOME, ID FORNITURA E DATA FORNITURA

--DELLE FORNITURE SPEDITE DURANTE GLI ULTIMI 3 MESI

```
CREATE VIEW ULTIME_FORN AS
SELECT FORNITORE.NOME_FORNITORE, FORNITORE.COGNOME_FORNITORE,
FORNITURA.ID_FORNITURA, FORNITURA.DATA_FORNITURA
FROM FORNITURA JOIN FORNITORE ON FORNITURA.AZIENDA_FORNITORE =
FORNITORE.AZIENDA_FORNITORE
WHERE (SYSDATE - FORNITURA.DATA_FORNITURA) < 90
```

User: MANAGER_LOGISTICA

--VISTA CHE VISUALIZZA ID FORNITURA, ID PRODOTTO, TAGLIA FORNITA E PREZZO DELLA FORNITURA

--PIU' COSTOSA

```
CREATE VIEW FORNITURA_MAX AS
SELECT ID_FORNITURA, ID_PRODOTTO, TAGLIA_FORNITA ,
PREZZO_FORNITURA
FROM FORNITURA
WHERE PREZZO_FORNITURA = (SELECT MAX(PREZZO_FORNITURA) FROM
FORNITURA)
```

User: MANAGER_LOGISTICA

--VISTA CHE VISUALIZZA ID FORNITURA, ID PRODOTTO, TAGLIA FORNITA E PREZZO DELLA FORNITURA

--MENO COSTOSA

```
CREATE VIEW FORNITURA_MIN AS
SELECT ID_FORNITURA, ID_PRODOTTO, TAGLIA_FORNITA ,
PREZZO_FORNITURA
FROM FORNITURA
WHERE PREZZO_FORNITURA = (SELECT MIN(PREZZO_FORNITURA) FROM
FORNITURA)
```

User: MANAGER_SERVIZI

--VISTA CHE VISUALIZZA TUTTI I NOMI DEI CATALOGHI CON FASCIA
D'ETA' ADULTO

```
CREATE VIEW CATALOGHI AS  
SELECT NOME_CATALOGO, FASCIA_D_ETA  
FROM CATALOGO  
WHERE FASCIA_D_ETA = 'ADULTO'
```

User: MANAGER_SERVIZI

--VISTA CHE VISUALIZZA NOME, DATA E TIPO DI RELEASE CHE SI
TERRANNO SUCCESSIVAMENTE
--ALLA DATA DI SISTEMA

```
CREATE VIEW LAST_RELEASE AS  
SELECT NOME_RELEASE, DATA_RELEASE, TIPO_RELEASE  
FROM RELEASE_DAY  
WHERE DATA_RELEASE > SYSDATE
```

CAPITOLO 11

Conclusione

Il progetto termina qui, tuttavia in una situazione reale, ci sarebbe ancora una cosa da fare.

Un'interfaccia grafica per l'utente da poter utilizzare sfruttando: caselle di testo, bottoni, e altri elementi, che potrebbero rendere più semplice l'uso di quella che era al tempo la nostra idea di database.

Spunti e riferimenti

Per la realizzazione della base di dati abbiamo preso spunto da vari siti, affinché il risultato della nostra base di dati sia più realistica e conforme al mercato reale.

Per fare ciò siamo partiti da un'idea precisa, partendo dal voler costruire una base di dati per un sito di reselling online; di seguito quindi verrà linkato il sito che ha costituito le fondamenta delle nostre idee, essendo il miglior sito nell'ambito in cui ci siamo addentrati, cioè 'Stockx' :

<https://stockx.com/>

Per la gestione delle entità che abbiamo riportato, abbiamo cercato ogni modo possibile di renderle realistiche e provare a caratterizzare la nostra base di dati.

Per quanto concerne la maggior parte delle entità ci siamo quindi basati su Stockx, sviluppando da un modello di base delle nostre idee, come per: il Release Day, le Promo...invece per quanto riguarda l'entità catalogo abbiamo preso spunto da Zalando, che ha diversi cataloghi a disposizione, e abbiamo provato a rendere quell'idea il più possibile nostra.

Di seguito verrà riportato il link di Zalando, affinché, come nel caso di Stockx, venga citata la fonte:

<https://www.zalando.it/>

Ringraziamenti

Siamo quindi giunti al termine del nostro progetto per l'A.A 2021/2022, ringraziamo quindi il professore per averci dato l'opportunità di poterci migliorare e di esperire, tramite questo progetto, un argomento di interesse comune tra noi progettisti, ampliando ancora di più anche le conoscenze sull'argomento trattato; sperando magari che possa essere anche un buon prototipo da cui ripartire una volta terminati gli studi.