Giulio Giusteschi 20230A (Informatica)

Documentazione Progetto di Basi di Dati:

[Schema ER 3](#_Toc202805404)

[Schema Logico 4](#_Toc202805405)

[Relazioni e Attributi 4](#_Toc202805406)

[Chiavi Esterne e Riferimenti 4](#_Toc202805407)

[Funzioni 5](#_Toc202805408)

[Autenticazione 5](#_Toc202805409)

[login\_utente 5](#_Toc202805410)

[modifica\_password 5](#_Toc202805411)

[Gestione Utenti 5](#_Toc202805412)

[crea\_cliente 5](#_Toc202805413)

[elimina\_cliente 6](#_Toc202805414)

[modifica\_dato\_cliente 6](#_Toc202805415)

[Gestione Negozi 6](#_Toc202805416)

[crea\_negozio 6](#_Toc202805417)

[elimina\_negozio 7](#_Toc202805418)

[modifica\_negozio 7](#_Toc202805419)

[Gestione Prodotti 7](#_Toc202805420)

[crea\_prodotto 7](#_Toc202805421)

[elimina\_prodotto 8](#_Toc202805422)

[modifica\_prodotto 8](#_Toc202805423)

[Prodotti nei Negozi 8](#_Toc202805424)

[aggiungi\_vendita\_prodotto 8](#_Toc202805425)

[elimina\_vendita\_prodotto 8](#_Toc202805426)

[modifica\_vendita\_prodotto 9](#_Toc202805427)

[Gestione Fornitori 9](#_Toc202805428)

[crea\_fornitore 9](#_Toc202805429)

[elimina\_fornitore 9](#_Toc202805430)

[modifica\_fornitore 10](#_Toc202805431)

[Gestione Forniture 10](#_Toc202805432)

[aggiungi\_fornitura 10](#_Toc202805433)

[elimina\_fornitura 10](#_Toc202805434)

[modifica\_fornitura 11](#_Toc202805435)

[Cliente – Visualizzazione e Acquisto 11](#_Toc202805436)

[prodotti\_disponibili\_negozio 11](#_Toc202805437)

[visualizza\_saldo\_punti 11](#_Toc202805438)

[acquisto\_cliente 11](#_Toc202805439)

[effettua\_acquisto 12](#_Toc202805440)

[Ordini per negozi 12](#_Toc202805441)

[effettua\_ordine 12](#_Toc202805442)

[Trigger 13](#_Toc202805443)

[trigger\_aggiorna\_punti 13](#_Toc202805444)

[trigger\_aggiorna\_disponibilita 13](#_Toc202805445)

[trigger\_quantita\_zero 13](#_Toc202805446)

[Viste 14](#_Toc202805447)

[ListaTesserati 14](#_Toc202805448)

[StoricoOrdiniFornitori 14](#_Toc202805449)

[ClientiConPiuDi300Punti 14](#_Toc202805450)

[StoricoTessere 14](#_Toc202805451)

[Scipt PHP 14](#_Toc202805452)

[Autenticazione e sessione 14](#_Toc202805453)

[login.php 14](#_Toc202805454)

[logout.php 14](#_Toc202805455)

[modifica\_password.php 15](#_Toc202805456)

[db.php 15](#_Toc202805457)

[Gestione Clienti 15](#_Toc202805458)

[gestione\_clienti.php 15](#_Toc202805459)

[Acquisti (cliente) 15](#_Toc202805460)

[acquisti.php 15](#_Toc202805461)

[checkout.php 16](#_Toc202805462)

[conferma\_acquisto.php 16](#_Toc202805463)

[Gestione Negozi 16](#_Toc202805464)

[gestione\_negozi.php 16](#_Toc202805465)

[info\_negozio.php 17](#_Toc202805466)

[negozi\_chiusi.php 18](#_Toc202805467)

[Dashboards 19](#_Toc202805468)

[dashboard\_cliente.php 19](#_Toc202805469)

[dashboard\_manager.php 19](#_Toc202805470)

[Gestione Prodotti 20](#_Toc202805471)

[gestione\_prodotti.php 20](#_Toc202805472)

[Ordini e Forniture 20](#_Toc202805473)

[ordini\_fornitore.php 20](#_Toc202805474)

[effettua\_ordine.php 21](#_Toc202805475)

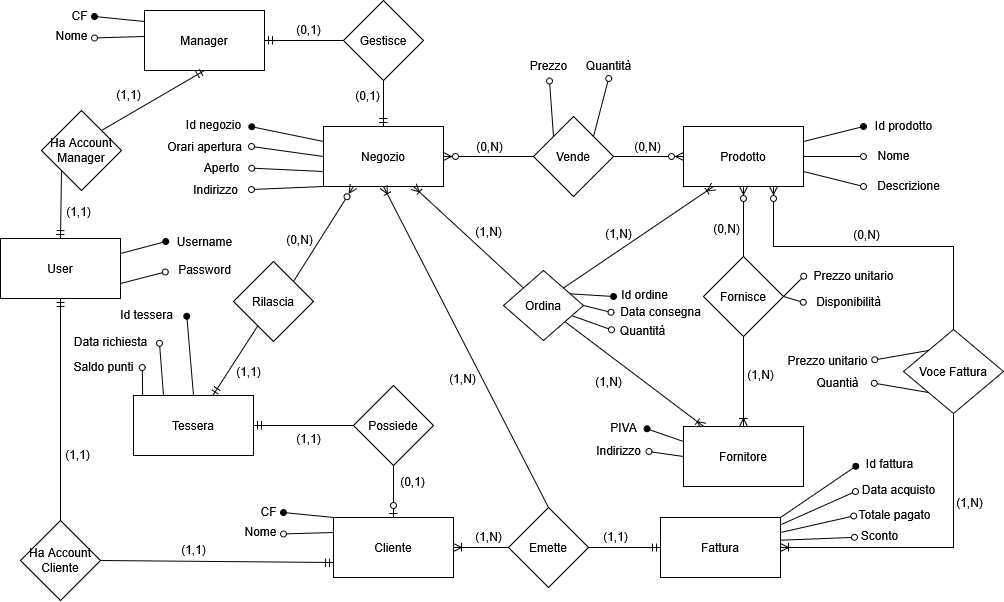
[Fatture 21](#_Toc202805476)

[fatture.php 21](#_Toc202805477)

[Tessere +300p 22](#_Toc202805478)

[saldi\_punti.php 22](#_Toc202805479)

# Schema ER



# Schema Logico

## Relazioni e Attributi

|  |  |
| --- | --- |
| Relazione | Attributi |
| Negozio | IDNegozio (PK), ManagerCF (FK), OrariApertura, Indirizzo, Aperto |
| Vende | NegozioID (FK), ProdottoID (FK) (PK = entrambi), Prezzo, Quantita |
| Prodotto | IDProdotto (PK), Nome, Descrizione |
| Fornisce | FornitorePIVA (FK), ProdottoID (FK) (PK = entrambi), PrezzoUnitario, Disponibilita |
| Fornitore | PIVA (PK), Indirizzo |
| Ordina | IDOrdine (PK), NegozioID (FK), ProdottoID (FK), FornitorePIVA (FK), DataConsegna, Quantita |
| Fattura | IDFattura (PK), ClienteCF (FK), NegozioID (FK), DataAcquisto, ScontoApplicato, TotalePagato |
| VoceFattura | FatturaID (FK), ProdottoID (FK) (PK = entrambi), PrezzoUnitario, Quantita |
| Tessera | IDTessera (PK), DataRichiesta, NegozioID (FK), ClienteCF (FK), SaldoPunti |
| Cliente | CF (PK), Nome, Username (FK) |
| Manager | CF (PK), Nome, Username (FK) |
| User | Username (PK), Password |

## Chiavi Esterne e Riferimenti

|  |  |
| --- | --- |
| Chiave Esterna | Referenza |
| Negozio.ManagerCF | Manager.CF |
| Vende.NegozioID | Negozio.IDNegozio |
| Vende.ProdottoID | Prodotto.IDProdotto |
| Fornisce.FornitorePIVA | Fornitore.PIVA |
| Fornisce.ProdottoID | Prodotto.IDProdotto |
| Ordina.NegozioID | Negozio.IDNegozio |
| Ordina.ProdottoID | Prodotto.IDProdotto |
| Ordina.FornitorePIVA | Fornitore.PIVA |
| Fattura.ClienteCF | Cliente.CF |
| Fattura.NegozioID | Negozio.IDNegozio |
| VoceFattura.FatturaID | Fattura.IDFattura |
| VoceFattura.ProdottoID | Prodotto.IDProdotto |
| Tessera.NegozioID | Negozio.IDNegozio |
| Tessera.ClienteCF | Cliente.CF |
| Cliente.Username | User.Username |
| Manager.Username | User.Username |

# Funzioni

## Autenticazione

### login\_utente

FUNCTION login\_utente(p\_username VARCHAR, p\_password VARCHAR) RETURNS TABLE (username VARCHAR(50), codice\_fiscale VARCHAR(16), nome VARCHAR(100))

Funzione usata per autenticare gli utenti (manager e clienti).  
Accetta username e password e restituisce le informazioni dell’utente (username, codice fiscale, nome), se l’autenticazione ha successo.  
Verifica prima se l’utente è un cliente, se non trovato come cliente, verifica se è un manager.

### modifica\_password

FUNCTION modifica\_password(p\_username VARCHAR, p\_password\_corrente VARCHAR, p\_password\_nuova VARCHAR) RETURNS TEXT

Funzione utilizzata per consentire a clienti e manager di modificare la propria password.  
Controlla se l’utente esiste (cliente o manager), verifica che la password fornita sia corretta, se la nuova password è identica a quella attuale, infine aggiorna la password.  
Codici di stato in base all'esito dell'operazione:

* "USER\_NOT\_FOUND" se l’utente non esiste (né come cliente né come manager)
* "WRONG\_PASSWORD" se la password corrente fornita non è corretta
* "SAME\_PASSWORD" se la nuova password coincide con quella attuale
* "OK" se la modifica della password è avvenuta con successo

## Gestione Utenti

### crea\_cliente

FUNCTION crea\_cliente(p\_cf VARCHAR, p\_nome VARCHAR, p\_username VARCHAR) RETURNS TEXT

Funzione utilizzata dal manager per creare un nuovo cliente.  
Verifica l’unicità del codice fiscale e dello username forniti. Se non esistono conflitti, crea prima l’utente con password predefinita "1234", poi crea l’istanza del cliente associata.  
Codici di stato in base all'esito dell'operazione:

* "USERNAME\_DUPLICATO" se lo username è già associato a un altro utente
* "CF\_DUPLICATO" se il codice fiscale è già associato a un altro cliente
* "OK" se la creazione del cliente è avvenuta con successo

### elimina\_cliente

FUNCTION elimina\_cliente(p\_cf VARCHAR) RETURNS TEXT

Funzione utilizzata dal manager per eliminare un cliente.  
Accetta un codice fiscale, individua l’username associato e rimuove il cliente. Grazie al CASCADE, vengono eliminati automaticamente anche l’utente e l’eventuale tessera fedeltà associata.  
Codici di stato in base all'esito dell'operazione:

* "CLIENTE\_NOT\_FOUND" se non esiste alcun cliente con il codice fiscale specificato
* "OK" se il cliente è stato eliminato con successo

### modifica\_dato\_cliente

FUNCTION modifica\_dato\_cliente(p\_cf\_attuale VARCHAR, p\_campo TEXT, p\_valore\_nuovo TEXT) RETURNS TEXT

Funzione utilizzata dal manager per modificare i dati di un cliente.  
Consente di modificare nome, codice fiscale o username del cliente.   
Controlla che il cliente esista, se si modifica il codice fiscale o lo username, controlla che non ci siano duplicati, e In caso di modifica dello username, aggiorna sia la tabella Cliente che Utente per mantenere la coerenza.  
Codici di stato in base all'esito dell'operazione:

* "CLIENTE\_NOT\_FOUND" se il cliente non esiste
* "CF\_DUPLICATO" se il nuovo codice fiscale è già assegnato a un altro cliente
* "USERNAME\_DUPLICATO" se il nuovo username è già usato da un altro utente
* "CAMPO\_NON\_VALIDO" se il campo fornito non è gestito dalla funzione
* "OK" se la modifica è stata eseguita correttamente

## Gestione Negozi

### crea\_negozio

FUNCTION crea\_negozio(p\_orari VARCHAR, p\_indirizzo TEXT, p\_cf\_manager VARCHAR DEFAULT NULL) RETURNS TEXT

Funzione utilizzata dal manager per creare un nuovo negozio.  
Consente di specificare gli orari di apertura, l’indirizzo e facoltativamente il codice fiscale del manager responsabile. Se il manager è specificato, viene verificata la sua esistenza.  
Codici di stato in base all'esito dell'operazione:

* "MANAGER\_NOT\_FOUND" se il codice fiscale specificato non corrisponde a nessun manager
* "OK" se il negozio è stato creato con successo

### elimina\_negozio

FUNCTION elimina\_negozio(p\_id INTEGER) RETURNS TEXT

Funzione utilizzata per chiudere un negozio esistente.  
Dato l’ID di un negozio, ne imposta l’attributo “Aperto” a FALSE e cancella gli orari di apertura. Il negozio non viene fisicamente eliminato, ma reso inattivo.  
Codici di stato in base all'esito dell'operazione:

* "NEGOZIO\_NOT\_FOUND" se il negozio specificato non esiste
* "OK" se il negozio è stato disattivato correttamente

### modifica\_negozio

FUNCTION modifica\_negozio(p\_id INTEGER, p\_campo TEXT, p\_valore TEXT) RETURNS TEXT

Funzione utilizzata per modificare le informazioni di un negozio.  
È possibile aggiornare gli orari di apertura, l’indirizzo o il manager associato. Se il campo da modificare è il manager, viene verificata la sua esistenza nel sistema.  
Codici di stato in base all'esito dell'operazione:

* "NEGOZIO\_NOT\_FOUND" se il negozio specificato non esiste
* "MANAGER\_NOT\_FOUND" se il nuovo manager specificato non esiste
* "CAMPO\_NON\_VALIDO" se il campo da modificare non è presente nella funzione
* "OK" se la modifica è stata effettuata con successo

## Gestione Prodotti

### crea\_prodotto

FUNCTION crea\_prodotto(p\_nome VARCHAR, p\_descrizione TEXT) RETURNS TEXT

Funzione utilizzata per creare un nuovo prodotto.  
Accetta nome e descrizione del prodotto, verifica che il nome non sia già presente nel database e, se non ci sono duplicati, inserisce il nuovo prodotto.  
Codici di stato in base all'esito dell'operazione:

* "NOME\_DUPLICATO" se esiste già un prodotto con lo stesso nome
* "OK" se il prodotto è stato creato con successo

### elimina\_prodotto

FUNCTION elimina\_prodotto(p\_id INTEGER) RETURNS TEXT

Funzione utilizzata per eliminare un prodotto tramite il suo ID.  
Se il prodotto esiste, viene rimosso dal database.  
Codici di stato in base all'esito dell'operazione:

* "PRODOTTO\_NOT\_FOUND" se il prodotto specificato non esiste
* "OK" se il prodotto è stato eliminato correttamente

### modifica\_prodotto

FUNCTION modifica\_prodotto(p\_id INTEGER, p\_campo TEXT, p\_valore TEXT) RETURNS TEXT

Funzione utilizzata per modificare i dati di un prodotto.  
Permette di aggiornare il nome o la descrizione di un prodotto esistente.  
Codici di stato in base all'esito dell'operazione:

* "PRODOTTO\_NOT\_FOUND" se il prodotto specificato non esiste
* "CAMPO\_NON\_VALIDO" se il campo fornito non è supportato dalla funzione
* "OK" se la modifica è stata eseguita correttamente

## Prodotti nei Negozi

### aggiungi\_vendita\_prodotto

FUNCTION aggiungi\_vendita\_prodotto(p\_id\_negozio INTEGER, p\_id\_prodotto INTEGER, p\_prezzo NUMERIC, p\_quantita INTEGER) RETURNS TEXT

Funzione utilizzata per aggiungere un nuovo prodotto in vendita presso un negozio.  
Verifica che il negozio e il prodotto esistano, e che la combinazione negozio-prodotto non sia già presente. Se tutti i controlli sono superati, inserisce il prodotto con il relativo prezzo e quantità.  
Codici di stato in base all'esito dell'operazione:

* "NEGOZIO\_NOT\_FOUND" se il negozio non esiste
* "PRODOTTO\_NOT\_FOUND" se il prodotto non esiste
* "PRODOTTO\_GIA\_PRESENTE" se il prodotto è già in vendita nel negozio
* "OK" se l’inserimento è avvenuto con successo

### elimina\_vendita\_prodotto

FUNCTION elimina\_vendita\_prodotto(p\_id\_negozio INTEGER, p\_id\_prodotto INTEGER) RETURNS TEXT

Funzione utilizzata per rimuovere un prodotto dalla lista dei prodotti in vendita presso un negozio.  
Verifica che la combinazione negozio-prodotto esista e, in tal caso, elimina la voce corrispondente.  
Codici di stato in base all'esito dell'operazione:

* "VOCE\_NON\_TROVATA" se il prodotto non è in vendita nel negozio specificato
* "OK" se il prodotto è stato rimosso correttamente

### modifica\_vendita\_prodotto

FUNCTION modifica\_vendita\_prodotto(p\_id\_negozio INTEGER, p\_id\_prodotto INTEGER, p\_campo TEXT, p\_valore NUMERIC) RETURNS TEXT

Funzione utilizzata per modificare il prezzo o la quantità di un prodotto venduto in un negozio.  
Verifica che il prodotto sia effettivamente venduto nel negozio e aggiorna il campo specificato.  
Codici di stato in base all'esito dell'operazione:

* "VOCE\_NON\_TROVATA" se il prodotto non è in vendita nel negozio specificato
* "CAMPO\_NON\_VALIDO" se il campo specificato non è un campo accettato dalla funzione
* "OK" se la modifica è stata eseguita correttamente

## Gestione Fornitori

### crea\_fornitore

FUNCTION crea\_fornitore(p\_piva VARCHAR, p\_indirizzo TEXT) RETURNS TEXT

Funzione utilizzata per creare un nuovo fornitore.  
Accetta la partita IVA e l’indirizzo del fornitore. Verifica che la partita IVA non sia già presente nel database e se unica inserisce il nuovo fornitore.  
Codici di stato in base all'esito dell'operazione:

* "PIVA\_DUPLICATA" se esiste già un fornitore con la stessa partita IVA
* "OK" se il fornitore è stato creato correttamente

### elimina\_fornitore

FUNCTION elimina\_fornitore(p\_piva VARCHAR) RETURNS TEXT

Funzione utilizzata per eliminare un fornitore dal database.  
Accetta la partita IVA del fornitore e, se il fornitore esiste, lo rimuove.  
Codici di stato in base all'esito dell'operazione:

* "FORNITORE\_NOT\_FOUND" se il fornitore specificato non esiste
* "OK" se il fornitore è stato eliminato con successo

### modifica\_fornitore

FUNCTION modifica\_fornitore(p\_piva\_attuale VARCHAR, p\_campo TEXT, p\_valore TEXT) RETURNS TEXT

Funzione utilizzata per modificare le informazioni di un fornitore.  
Permette di aggiornare l’indirizzo o la partita IVA di un fornitore esistente. In caso di aggiornamento della partita IVA, controlla che non esistano duplicati.  
Codici di stato in base all'esito dell'operazione:

* "FORNITORE\_NOT\_FOUND" se il fornitore specificato non esiste
* "PIVA\_DUPLICATA" se la nuova partita IVA è già assegnata a un altro fornitore
* "CAMPO\_NON\_VALIDO" se il campo specificato non è supportato dalla funzione
* "OK" se la modifica è stata eseguita con successo

## Gestione Forniture

### aggiungi\_fornitura

FUNCTION aggiungi\_fornitura(p\_piva VARCHAR, p\_prodotto INTEGER, p\_prezzo NUMERIC, p\_disp INTEGER) RETURNS TEXT

Funzione utilizzata per aggiungere una nuova fornitura di un prodotto da parte di un fornitore.  
Verifica che il fornitore e il prodotto esistano e che la combinazione non sia già presente. In tal caso, inserisce una nuova voce nella tabella Fornisce.  
Codici di stato in base all'esito dell'operazione:

* "FORNITORE\_NOT\_FOUND" se il fornitore specificato non esiste
* "PRODOTTO\_NOT\_FOUND" se il prodotto specificato non esiste
* "FORNITURA\_DUPLICATA" se esiste già una fornitura per quel prodotto da parte del fornitore
* "OK" se la fornitura è stata inserita correttamente

### elimina\_fornitura

FUNCTION elimina\_fornitura(p\_piva VARCHAR, p\_prodotto INTEGER) RETURNS TEXT

Funzione utilizzata per eliminare una fornitura esistente.  
Verifica che la combinazione fornitore-prodotto esista e, se presente, la rimuove dalla tabella Fornisce.  
Codici di stato in base all'esito dell'operazione:

* "FORNITURA\_NON\_TROVATA" se la fornitura specificata non esiste
* "OK" se la fornitura è stata eliminata con successo

### modifica\_fornitura

FUNCTION modifica\_fornitura(p\_piva VARCHAR, p\_prodotto INTEGER, p\_campo TEXT, p\_valore NUMERIC) RETURNS TEXT

Funzione utilizzata per modificare i dati di una fornitura esistente.  
Consente di aggiornare il prezzo unitario o la disponibilità del prodotto da parte del fornitore. Verifica la presenza della fornitura e il campo specificato.  
Codici di stato in base all'esito dell'operazione:

* "FORNITURA\_NON\_TROVATA" se la combinazione specificata non esiste
* "CAMPO\_NON\_VALIDO" se il campo specificato non è supportato dalla funzione
* "OK" se la modifica è stata effettuata con successo

## Cliente – Visualizzazione e Acquisto

### prodotti\_disponibili\_negozio

FUNCTION prodotti\_disponibili\_negozio(p\_id\_negozio INTEGER) RETURNS TABLE (id\_prodotto INTEGER, nome\_prodotto VARCHAR, descrizione TEXT, prezzo NUMERIC, quantita INTEGER)

Funzione utilizzata per restituire tutti i prodotti disponibili in un determinato negozio.  
Ritorna solo i prodotti con quantità maggiore di zero, mostrando le informazioni dettagliate di ciascun prodotto (nome, descrizione, prezzo, quantità).

### visualizza\_saldo\_punti

FUNCTION visualizza\_saldo\_punti(p\_cf\_cliente VARCHAR) RETURNS TABLE (id\_tessera INTEGER, data\_richiesta DATE, negozio\_id INTEGER, saldo\_punti INTEGER)

Funzione utilizzata per mostrare le informazioni relative alla tessera fedeltà di un cliente.  
Dato il codice fiscale del cliente, restituisce ID tessera, data di richiesta, negozio emittente e saldo attuale dei punti.

### acquisto\_cliente

FUNCTION acquisto\_cliente(p\_CF\_cliente VARCHAR, p\_ID\_negozio INTEGER, p\_importo NUMERIC, p\_sconto BOOLEAN) RETURNS TEXT

Funzione utilizzata per registrare una nuova fattura e applicare eventuali sconti tramite tessera.  
Se il cliente seleziona l’opzione sconto gli verrà in automatico applicato il massimo sconto disponibile in base al saldo punti del cliente. Se applicato, lo sconto non può superare i 100 euro. I punti utilizzati vengono scalati dal saldo. Se l’utente ha selezionato sconto senza poter riscattare nemmeno il più basso, l’acquisto avverrà comunque senza applicare alcuno sconto alla fattura.  
Codici di stato in base all'esito dell'operazione:

* "OK\_SCONTO\_5\_<ID>" / "OK\_SCONTO\_15\_<ID>" / "OK\_SCONTO\_30\_<ID>" se viene applicato uno sconto e creata la fattura
* "OK\_NO\_SCONTO" se non ci sono abbastanza punti per uno sconto
* "OK\_SENZA\_RICHIESTA" se il cliente non ha richiesto lo sconto

### effettua\_acquisto

FUNCTION effettua\_acquisto(p\_CF\_cliente VARCHAR, p\_ID\_negozio INTEGER, p\_prodotti INTEGER[], p\_quantita INTEGER[], p\_applica\_sconto BOOLEAN) RETURNS TEXT

Funzione usata per completare l’acquisto di uno o più prodotti in un negozio.  
Effettua tutti i controlli sulle quantità disponibili, calcola il totale e invoca la funzione `acquisto\_cliente`. Successivamente registra ogni voce in VoceFattura e aggiorna le quantità nel magazzino del negozio.  
Codici di stato in base all'esito dell'operazione:

* "ARRAY\_LENGTH\_MISMATCH" se gli array di prodotti e quantità hanno lunghezza diversa
* "PRODOTTO\_NON\_PRESENTE" se un prodotto non è venduto nel negozio selezionato
* "QUANTITA\_NON\_DISPONIBILE" se non c’è abbastanza disponibilità per almeno un prodotto
* "OK\_\*\_<ID>" se l’acquisto è andato a buon fine (con o senza sconto)

## Ordini per negozi

### effettua\_ordine

FUNCTION effettua\_ordine(p\_ID\_negozio INTEGER, p\_ID\_prodotto INTEGER, p\_quantita INTEGER) RETURNS TEXT

Funzione utilizzata per effettuare un ordine di rifornimento da parte di un negozio.  
Verifica che il negozio esista e sia aperto. Se ci sono fornitori con disponibilità sufficiente del prodotto, sceglie automaticamente quello con il prezzo più basso e registra l’ordine nella tabella Ordina.  
Codici di stato in base all'esito dell'operazione:

* "NEGOZIO\_NOT\_FOUND" se il negozio specificato non esiste
* "NEGOZIO\_CHIUSO" se il negozio esiste ma non è aperto
* "NESSUN\_FORNITORE\_DISPONIBILE" se nessun fornitore ha disponibilità sufficiente del prodotto
* "OK" se l’ordine è stato effettuato con successo

# Trigger

### trigger\_aggiorna\_punti

TRIGGER trigger\_aggiorna\_punti AFTER INSERT ON Fattura FOR EACH ROW EXECUTE FUNCTION aggiorna\_punti\_tessera()

Trigger utilizzato per aggiornare automaticamente il saldo punti della tessera del cliente ogni volta che viene inserita una nuova fattura.  
Richiama la funzione aggiorna\_punti\_tessera() che aggiunge al saldo un punto per ogni euro speso.

### trigger\_aggiorna\_disponibilita

TRIGGER trigger\_aggiorna\_disponibilita AFTER INSERT ON Ordina FOR EACH ROW EXECUTE FUNCTION aggiorna\_disponibilita\_fornitore()

Trigger utilizzato per aggiornare la disponibilità di un prodotto presso un fornitore ogni volta che viene inserito un nuovo ordine.  
Richiama la funzione aggiorna\_disponibilita\_fornitore() che scala automaticamente la quantità ordinata.

### trigger\_quantita\_zero

TRIGGER trigger\_quantita\_zero AFTER UPDATE ON Vende FOR EACH ROW WHEN (NEW.Quantita = 0 AND OLD.Quantita > 0) EXECUTE FUNCTION gestisci\_quantita\_zero()

Trigger utilizzato per rimuovere un prodotto da un negozio quando la quantità scende a zero e per avviare automaticamente un nuovo ordine di rifornimento.  
Richiama la funzione gestisci\_quantita\_zero() che elimina la riga da Vende e utilizza effettua\_ordine per ordinare 1 singola unità.

# Viste

### ListaTesserati

VIEW ListaTesserati

Vista che mostra la lista dei clienti che possiedono una tessera fedeltà, indicando negozio, dati cliente, data di richiesta e saldo punti.

### StoricoOrdiniFornitori

VIEW StoricoOrdiniFornitori

Vista che raccoglie tutti gli ordini effettuati dai negozi verso i fornitori, mostrando i dettagli dell’ordine, del prodotto e del fornitore.

### ClientiConPiuDi300Punti

VIEW ClientiConPiuDi300Punti

Vista che elenca tutti i clienti con una tessera fedeltà il cui saldo punti supera 300.

### StoricoTessere

VIEW StoricoTessere

Vista che mostra tutte le tessere emesse da negozi attualmente chiusi, includendo ID tessera, indirizzo del negozio e data di richiesta.

# Scipt PHP

## Autenticazione e sessione

### Immagine che contiene testo, schermata, Carattere, numero Il contenuto generato dall'IA potrebbe non essere corretto.login.php

Script utilizzato per gestire l'autenticazione dei clienti e dei manager. Riceve username e password tramite POST e invoca la funzione SQL login\_utente. Se l'autenticazione ha successo, avvia una sessione e reindirizza l'utente alla dashboard corretta. In caso contrario, mostra un messaggio di errore.

### logout.php

Script utilizzato per terminare la sessione utente. Distrugge tutti i dati della sessione e reindirizza l’utente alla pagina di login.

### modifica\_password.php

Script che consente a un utente autenticato (cliente o manager) di modificare la propria password.  
Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.Viene richiesto l’inserimento della password attuale e di quella nuova due volte. La validazione avviene tramite la funzione SQL modifica\_password.

### db.php

Script responsabile della connessione al database PostgreSQL.  
Utilizza la funzione pg\_connect con una stringa di connessione e rende disponibile la variabile globale $conn, utilizzata da tutti gli altri script per eseguire query sul database.

## Gestione Clienti

### gestione\_clienti.php

Script utilizzato dal manager per gestire i dati dei clienti.  
Consente di creare, modificare e eliminare clienti dal sistema. Le operazioni avvengono tramite form HTML e i dati vengono validati e passati alle rispettive funzioni SQL: crea\_cliente, modifica\_cliente, elimina\_cliente. Immagine che contiene testo, software, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

## Acquisti (cliente)

### acquisti.php

Script utilizzato dal cliente per visualizzare i prodotti disponibili in un negozio tramite l’apposita funzione SQL prodotti\_disponibili\_negozio.  
Consente la selezione dei prodotti da acquistare e l’aggiunta al carrello. Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

### checkout.php

Script che mostra il riepilogo del carrello del cliente. Permette di selezionare modificare le quantità dei prodotti selezionati e se applicare o meno lo sconto, infine invia i dati a conferma\_acquisto.php. Immagine che contiene testo, linea, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

### conferma\_acquisto.php

Script che completa l'acquisto di uno o più prodotti. Riceve i dati dal carrello e invoca la funzione SQL effettua\_acquisto, che gestisce la fattura, lo sconto e l’aggiornamento del magazzino.

## Gestione Negozi

### gestione\_negozi.php

Script utilizzato dal manager per gestire i negozi.  
Consente di creare nuovi negozi, modificarne i dati (indirizzo, orari, manager) e chiuderli.

Inolte dà la possibilità di aggiungere, modificare ed eliminare prodotti dai negozi.

Le azioni sono eseguite tramite form HTML e richiamano le apposite funzioni SQL, crea\_negozio, modifica\_negozio, elimina\_negozio, aggiungi\_vendita\_prodotto, modifica\_vendita\_prodotto, elimina\_vendita\_prodotto. Immagine che contiene testo, schermata, numero, software

Il contenuto generato dall'IA potrebbe non essere corretto.

### info\_negozio.php

Script che consente al manager di visualizzare le info sul negozio a cui è assegnato.

Mostra informazioni generiche tipo ID, indirizzo, orari, numero prodotti disponibili, numero di tesserati presso il negozio, numero di fatture emesse e lo stato del negozio.

Inoltre, mostra anche i dati delle tessere e delle fatture emesse dal negozio.Immagine che contiene testo, schermata, numero, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

### negozi\_chiusi.php

Script che mostra l’elenco dei negozi chiusi.  
Per ogni negozio, consente la visualizzazione delle tessere emesse, delle fatture con le rispettive voci emesse prima che il negozio venisse chiuso.Immagine che contiene testo, schermata, numero, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

## Dashboards

### dashboard\_cliente.php

Pagina iniziale per il cliente dopo il login.

Mostra i dati del cliente, nome, codice fiscale, username, una finta password, con relativo pulsante per modificarla e il saldo attuale della tessera punti posseduta dal cliente. Immagine che contiene testo, schermata, linea, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

### dashboard\_manager.php

Pagina iniziale per il manager dopo il login.

Visualizza i dati del manager come nome, codice fiscale, username e una finta password con il relativo pulsante per la modifica.

Inoltre mostra alcuni dati sul negozio associato al manager come l’id e l’indirizzo.Immagine che contiene testo, linea, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

## Gestione Prodotti

### gestione\_prodotti.php

Script utilizzato dal manager per gestire i prodotti, i fornitori e le forniture.  
Permette la creazione, modifica e cancellazione di prodotti, fornitori e forniture tramite le apposite funzioni SQL, crea\_prodotto, modifica\_prodotto, elimina\_prodotto, crea\_fornitore, elimina\_fornitore, modifica\_fornitore, aggiungi\_fornitura, elimina\_fornitura, modifica\_fornitura. Immagine che contiene testo, schermata, software, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

## Ordini e Forniture

### ordini\_fornitore.php

Script che mostra lo storico di tutti gli ordini effettuati dai vari negozi verso i fornitori.  
Visualizza le informazioni sugli ordini, e permette con un pulsante andare a effettua\_ordine.php per poter effettuare un’ordine. Immagine che contiene testo, schermata, Carattere, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

### effettua\_ordine.php

Script utilizzato dal manager per eseguire manualmente un ordine di rifornimento di un prodotto presso un determinato negozio.  
Viene selezionato il negozio, il prodotto e la quantità, e viene invocata la funzione SQL effettua\_ordine per registrare l’operazione. Immagine che contiene testo, schermata, software

Il contenuto generato dall'IA potrebbe non essere corretto.

## Fatture

### fatture.php

Script utilizzato per visualizzare tutte le fatture effettuate da un cliente.  
Per ogni fattura, l’ID, la data, lo sconto, totale pagato e dettaglio dei prodotti acquistati.

Immagine che contiene testo, schermata, Carattere, bianco

Il contenuto generato dall'IA potrebbe non essere corretto.

## Tessere +300p

### saldi\_punti.php

Script utilizzato dal manager per visualizzare i dati dei clienti con più di 300 punti sulla propria tessera fedeltà. Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.