Università degli studi di Napoli Federico II

Corso di laurea in Informatica

Insegnamento di Object Orientation

Anno Accademico 2020/21

Progettazione e sviluppo di un sistema di gestione per un negozio di alimentari con supporto di un database di tipo PostgreSQL

Autori

Scarfato Francesco

Matricola N86003769

Cataldo Giuseppe

Matricola N86002288

Descrizione del Progetto

Il sistema realizzato permette di gestire all'utente il negozio di alimentari in maniera rapida e intuitiva.

Il progetto (così come l'interfaccia grafica) è stato diviso in tre macroaree:

- *Clienti*
- *Vendite*
- *Magazzino*

Ci si può spostare agevolmente tra queste aree tramite la ***Barra di navigazione*** sulla sinistra

Clienti è la macroarea riguardante la gestione dei dati del cliente e della tessera associata. Le funzionalità sviluppate per quest'area sono:

- Inserimento di nuovi clienti nella base di dati
- Controllo automatico sul formato di data
- Generazione automatica del Codice Fiscale
- Generazione automatica (tramite la base di dati) della tessera per il nuovo cliente
- Visualizzare le tessere con i relativi punti
- Visualizzare i dati anagrafici di un cliente
- Barra di ricerca per filtrare i risultati
- Possibilità di ordinamento crescente/decrescente
- Eliminazione di una specifica tessera

Magazzino è la macroarea che gestisce l'organizzazione delle scorte e del magazzino. Per questa area (e per tutte quelle inerenti all'inserimento o alla visualizzazione dei prodotti) è stata adottata un'ulteriore divisione, infatti i prodotti sono stati divisi in:

- Prodotti misurati in KG e il cui prezzo è €/kg
- Prodotti misurati in unità singole e il cui prezzo è €/unità

Della prima categoria fanno parte: Frutta, Farinacei, Latticini, Verdure

Della seconda categoria fanno parte: Confezionati, Uova

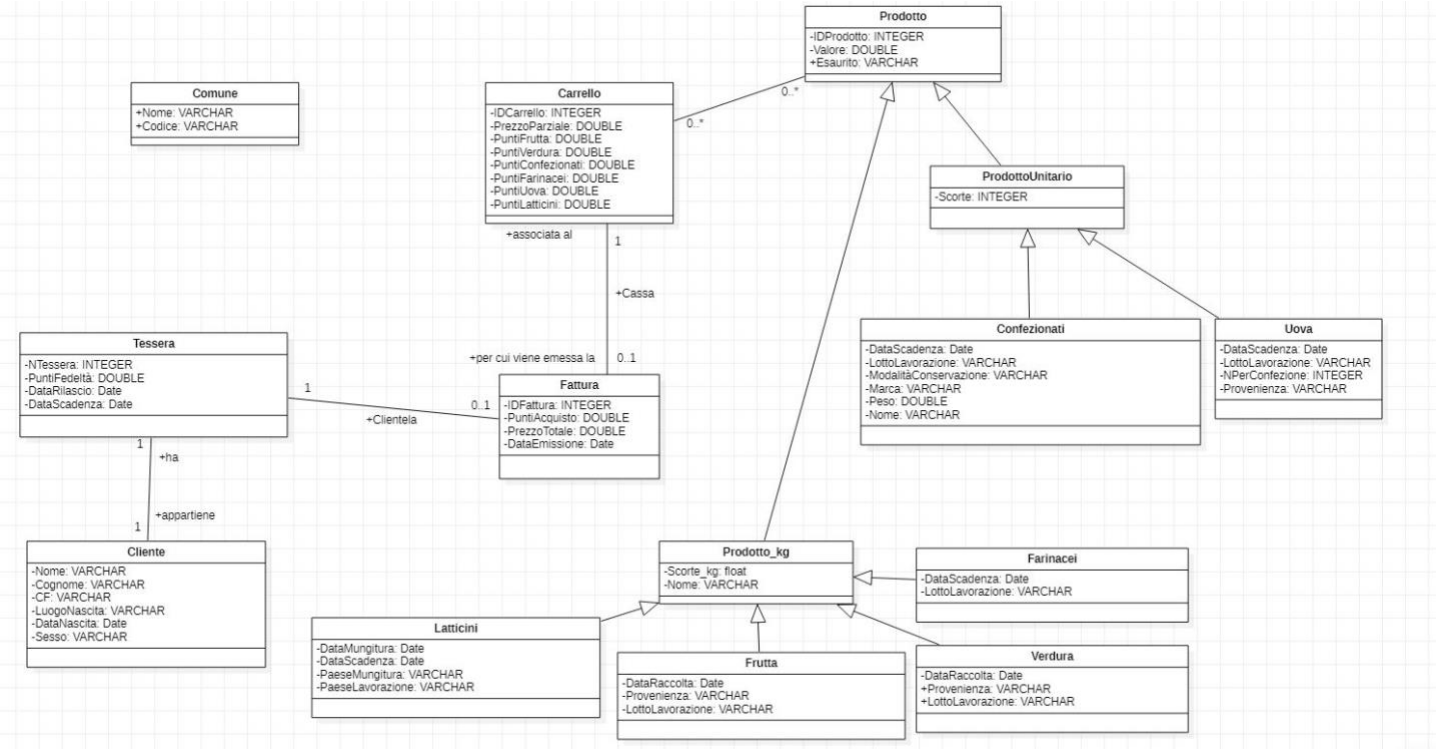
Le funzionalità sviluppate per quest'area sono:

- Inserimento di nuovi prodotti nella base di dati
- Controllo automatico sul formato di data
- Controllo automatico sulle incongruenze tra data corrente e data di scadenza/raccolta
- Visualizzare i vari tipi di prodotti inseriti
- Barra di ricerca per filtrare i risultati
- Possibilità di ordinamento crescente/decrescente
- Eliminazione di un prodotto specifico
- Gestione automatica dei prodotti scaduti
- Gestione automatica delle scorte esaurite

Vendite è la macroarea che gestisce l'interazione tra l'utente (venditore) e il cliente. Quest'area è a stretto contatto con l'area "Magazzino" che gestisce le scorte che saranno vendute. Le funzionalità sviluppate per quest'area sono:

- Visualizzare tutte le fatture emesse e le tessere alle quali sono associate
- Visualizzare in modo dettagliato una singola fattura che tiene conto dei punti totalizzati e del prezzo della singola vendita
- Creazione di un carrello in tempo reale
- Visualizzare e cercare prodotti per aggiungerli al carrello
- Gestione automatica dei punti per il singolo prodotto
- Gestione automatica del prezzo dipendente dalla quantità e dal valore del singolo prodotto
- Gestione automatica delle scorte (sia nel caso in cui vengano vendute sia nel caso in cui la vendita non venga finalizzata)
- Visualizzare i prodotti nel carrello
- Emissione fattura associata alla tessera del cliente

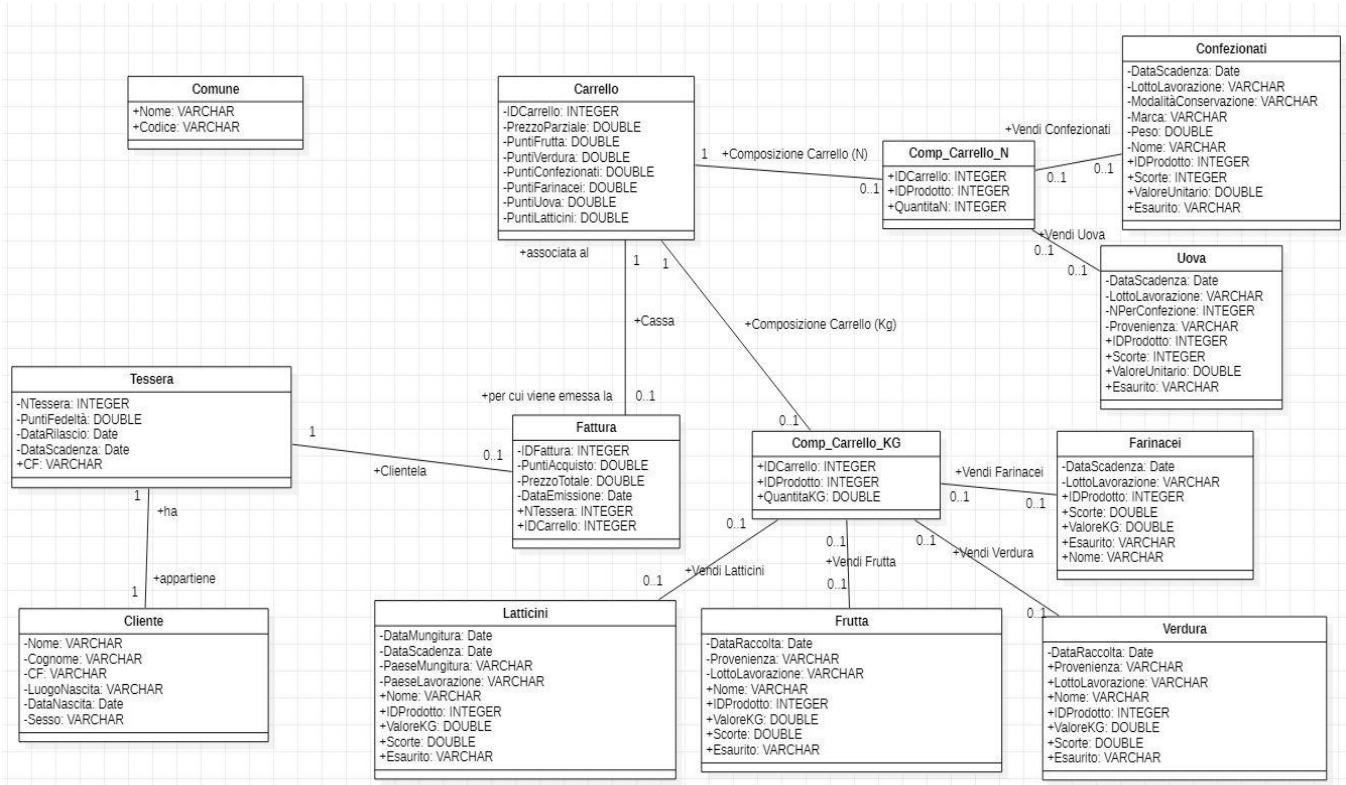
Class Diagram



Class Diagram Ristrutturato

Per ristrutturare il *Class Diagram* originale, vengono aggiunte alle varie classi le conseguenti *foreign key* per regolamentare le varie associazioni. La modifica più corposa riguarda i prodotti e il loro collegamento al carrello. Ciò avviene perché sono presenti due livelli di ereditarietà, infatti la classe “Prodotto” viene soppressa e gli attributi vengono passati alle classi “Prodotto Unitario” e “Prodotto KG”. A loro volta anche queste classi vengono eliminate e gli attributi vengono ereditati dalle classi sottostanti.

Vengono create due classi di intermezzo per collegare la classe “Carrello” e i vari prodotti poiché quest’ultima classe aveva un’associazione “molti a molti” con la classe “Prodotto”. Queste classi sono “Comp_Carrello_n” e “Comp_Carrello_kg”.



Dizionario delle classi

Classe	Descrizione
Comune	Classe per archiviare i nomi dei comuni e il loro codice per generare il codice fiscale
Cliente	Classe per archiviare i dati anagrafici di un cliente
Tessera	Classe per archiviare le tessere generate per ogni cliente registrato
Fattura	Classe per archiviare le fatture generate per ogni acquisto di ogni tessera
Carrello	Classe per registrare i prodotti che un cliente vuole acquistare, calcolare il prezzo totale e i punti per ogni categoria di prodotto
Comp_Carrello_n	Classe per indicare il numero di articoli (intero) che si vuole acquistare e in che carrello metterli
Comp_Carrello_KG	Classe per indicare la quantità di articoli che si vuole acquistare e in che carrello metterli
Confezionati	Classe per archiviare i dati di un prodotto di tipo Confezionato e le quantità di quel prodotto
Uova	Classe per archiviare i dati di un prodotto di tipo Uova e le quantità di quel prodotto
Farinacei	Classe per archiviare i dati di un prodotto di tipo Farinaceo e le quantità di quel prodotto
Verdura	Classe per archiviare i dati di un prodotto di tipo Verdura e le quantità di quel prodotto
Frutta	Classe per archiviare i dati di un prodotto di tipo Frutta e le quantità di quel prodotto
Latticini	Classe per archiviare i dati di un prodotto di tipo Latticino e le quantità di quel prodotto

Dizionario delle associazioni

Associazione	*Descrizione*	*Classi coinvolte*
Cliente - Tessera	Esprime la relazione del Cliente con la propria tessera personale	Cliente: ruolo “ha”, cardinalità 1 Tessera: ruolo “appartiene a”, cardinalità 1
Clientela	Esprime l’associazione delle varie Fatture del cliente e la sua Tessera	Tessera: cardinalità 1 Fattura: cardinalità 0..1
Cassa	Esprime la relazione tra il Carrello e la Fattura generata per quest’ultimo	Fattura: ruolo “associata al”, cardinalità 0..1 Carrello: ruolo “per cui viene emessa la”, cardinalità 1
Composizione Carrello (N)	Esprime la relazione tra il Carrello e Comp_Carrello_N quindi come vengono venduti i prodotti (n)	Carrello: cardinalità 1 Comp_Carrello_N: cardinalità 0..1
Composizione Carrello (Kg)	Esprime la relazione tra il Carrello e Comp_Carrello_Kg ovvero come vengono venduti i prodotti (Kg)	Carrello: cardinalità 1 Comp_Carrello_Kg: cardinalità 0..1
Vendi Latticini	Esprime la relazione tra Latticini e Comp_Carrello_Kg ovvero come vengono venduti i Latticini	Comp_Carrello_Kg: cardinalità 0..1 Latticini: cardinalità 0..1
Vendi Frutta	Esprime la relazione tra Frutta e Comp_Carrello_Kg ovvero come viene venduta la Frutta	Comp_Carrello_Kg: cardinalità 0..1 Frutta: cardinalità 0..1
Vendi Verdura	Esprime la relazione tra Verdura e Comp_Carrello_Kg ovvero come vengono vendute le Verdure	Comp_Carrello_Kg: cardinalità 0..1 Verdura: cardinalità 0..1
Vendi Farinacei	Esprime la relazione tra Farinacei e Comp_Carrello_Kg ovvero come vengono venduti i Farinacei	Comp_Carrello_Kg: cardinalità 0..1 Farinacei: cardinalità 0..1

Vendi Uova	Esprime la relazione tra Uova e Comp_Carrello_N ovvero come vengono venduti le Uova	Comp_Carrello_N: cardinalità 0..1 Uova: cardinalità 0..1
Vendi Confezionati	Esprime la relazione tra Confezionati e Comp_Carrello_N ovvero come vengono venduti i Confezionati	Comp_Carrello_N: cardinalità 0..1 Confezionati: cardinalità 0..1

Dizionario dei vincoli

Nome	**Descrizione**
carrello_pkey	Vincolo per identificare la Primary Key della tabella Carrello e quindi identificare univocamente ciascun carrello.
pk_cliente	Vincolo per identificare la Primary Key della tabella Cliente e quindi identificare univocamente ciascun cliente.
cliente_sesso_check	Vincolo per controllare che l'attributo sesso della classe Cliente sia correttamente inserito e quindi rientri nei due valori impostati in esso. (M, F)
nome_exp_reg	Vincolo per controllare che il valore attribuito all'attributo Nome della classe Cliente abbia un valore legittimo.
luogo_nascita_exp_reg	Vincolo per controllare che l'attributo Luogo di Nascita della classe Cliente abbia un valore legittimo.
cogn_exp_reg	Vincolo per controllare che l'attributo Cognome della classe Cliente abbia un valore legittimo.

cf_exp_reg	Vincolo per controllare che l'attributo CF della classe Cliente abbia un valore legittimo.
carrello_compkg_fk	Vincolo per identificare la Foreign Key della classe Comp_carrello_kg che si riferisce all'attributo id_carrello della classe Carrello
carello_compn_fk	Vincolo per identificare la Foreign Key della classe Comp_carrello_n che si riferisce all'attributo id_carrello della classe Carrello
confezionati_pkey	Vincolo per identificare la Primary Key della tabella Confezionati e quindi identificare univocamente ciascun prodotto confezionato.
confezionato_data_check	Vincolo per controllare che la data di scadenza inserita per ciascun confezionato sia maggiore della data nella quale è stato effettuato l'inserimento.
conf_exp_reg	Vincolo per controllare che l'attributo modalita_conservazione della classe Confezionato abbia un valore legittimo.
farinacei_pkey	Vincolo per identificare la Primary Key della tabella Farinacei e quindi identificare univocamente ciascun prodotto farinaceo.

farinaceo_data_check	Vincolo per controllare che la data di scadenza inserita per ciascun farinaceo sia maggiore della data nella quale è stato effettuato l'inserimento.
farin_exp_reg	Vincolo per controllare che il nome di ciascun farinaceo abbia un valore legittimo.
fattura_pkey	Vincolo per identificare la Primary Key della tabella Fattura e quindi identificare univocamente ciascuna fattura.
fattura_carrello_fk	Vincolo per identificare la Foreign Key della classe Fattura che si riferisce all'attributo id_carrello della classe Carrello.
fattura_tess_fk	Vincolo per identificare la Foreign Key della classe Fattura che si riferisce all'attributo n_tessera della classe Tessera.
frutta_pkey	Vincolo per identificare la Primary Key della tabella Frutta e quindi identificare univocamente ogni frutta.
frutta_data_check	Vincolo per controllare che la data di scadenza inserita per ogni

	frutta sia maggiore della data nella quale è stato effettuato l'inserimento.
frutta_exp_reg	Vincolo per controllare che il nome di ciascuna frutta abbia un valore legittimo.
latticini_pkey	Vincolo per identificare la Primary Key della tabella Latticini e quindi identificare univocamente ogni prodotto caseario.
latticino_data_check	Vincolo per controllare che la data di mungitura inserita per ogni prodotto caseario sia minore o uguale alla data nella quale è stato effettuato l'inserimento.
latticino_datascadenza_check	Vincolo per controllare che la data di scadenza inserita per ciascun prodotto caseario sia maggiore della data nella quale è stato effettuato l'inserimento.
latt_exp_reg	Vincolo per controllare che il paese di mungitura di ogni frutta abbia un valore legittimo.
date_latt_check	Vincolo per controllare che la data di scadenza inserita per ciascun prodotto caseario sia

	maggiore della data di mungitura inserita.
pk_tessera	Vincolo per identificare la Primary Key della tabella Tessera e quindi identificare univocamente ogni tessera.
tessera_cliente_fk	Vincolo per identificare la Foreign Key della classe Tessera che si riferisce all'attributo cf della classe Cliente.
uova_pkey	Vincolo per identificare la Primary Key della tabella Uova e quindi identificare univocamente ogni prodotto avicolo.
uova_datascadenza_check	Vincolo per controllare che la data di scadenza inserita per ciascun prodotto avicolo sia maggiore della data nella quale è stato effettuato l'inserimento.
uova_exp_reg	Vincolo per controllare che la provenienza di ciascun prodotto avicolo abbia un valore legittimo.
verdura_pkey	Vincolo per identificare la Primary Key della tabella Verdura e quindi identificare univocamente ogni verdura.

verdura_data_check	Vincolo per controllare che la data di raccolta inserita perogni verdura sia minore della data nella quale è stato effettuato l'inserimento.
verdura_exp_reg	Vincolo per controllare che la provenienza e il nome di ogni verdura abbia un valore legittimo.

Schema Logico

Comune	(Nome, Codice)
Cliente	(Nome, Cognome, CF, LuogoNascita, DataNascita, Sesso)
Tessera	(NTessera, PuntiFedeltà, DataRilascio, DataScadenza, CF) CF → Cliente
Fattura	(IDFattura, PuntiAcquisto, PrezzoTotale, DataEmissione, NTessera, IDCarrello) NTessera → Tessera IDCarrello → Carrello
Carrello	(IDCarrello, PrezzoParziale, PuntiFrutta, PuntiVerdura, PuntiConfezionati, PuntiFarinacei, PuntiUova, PuntiLatticini)
Comp_Carrello_Kg	(IDCarrello, IDProdotto, QuantitàKG) IDProdotto → Latticini, Frutta, Verdura, Farinacei IDCarrello → Carrello
Comp_Carrello_N	(IDCarrello, IDProdotto, QuantitàN) IDProdotto → Uova, Confezionati IDCarrello → Carrello

Latticini	(DataMungitura, DataScadenza, PaeseMungitura, PaeseLavorazione, Nome, IDProdotto, ValoreKG, Scorte, Esaurito)
Frutta	(DataRaccolta, Provenienza, LottoLavorazione, Nome, IDProdotto, ValoreKG, Scorte, Esaurito)
Verdura	(DataRaccolta, Provenienza, LottoLavorazione, Nome, IDProdotto, ValoreKG, Scorte, Esaurito)
Farinacei	(DataScadenza, LottoLavorazione, Nome, IDProdotto, ValoreKG, Scorte, Esaurito)
Uova	(DataScadenza, LottoLavorazione, NPerConfezione, Provenienza, IDProdotto, ValoreUnitario, Scorte, Esaurito)
Confezionati	(DataScadenza, LottoLavorazione, ModalitàConservazione, Marca, Peso, Nome, IDProdotto, ValoreUnitario, Scorte, Esaurito)

Descrizione funzioni implementate

Le funzioni implementate non vengono attivate direttamente da trigger specifici ma vengono chiamate direttamente da alcuni metodi delle classi Java.

gestisci_scadenze_confezionato () -> Elimina le istanze “Confezionato” in cui la data di scadenza è stata superata dalla data corrente

gestisci_scadenze_farinaceo () -> Elimina le istanze “Farinaceo” in cui la data di scadenza è stata superata dalla data corrente

gestisci_scadenze_latticino () -> Elimina le istanze “Latticino” in cui la data di scadenza è stata superata dalla data corrente

gestisci_scadenze_uova () -> Elimina le istanze “Uova” in cui la data di scadenza è stata superata dalla data corrente

Descrizione trigger implementati

I trigger implementati fanno uso di “trigger functions”.

NOME	*CLASSE*	*DESCRIZIONE*
auto_datascadenza	cliente	Genera la tessera al cliente appena inserito con una data di scadenza = data dell’inserimento + 5 anni
cancella_carrelloKG	comp_carrello_kg	Rimpingua le scorte (che vengono misurate in kg) quando un carrello viene cancellato e la vendita non viene effettuata. Pone l’attributo esaurito a NULL
check_venditaKG	comp_carrello_kg	Verifica se il prodotto (che viene misurato in kg) che vuole essere venduto è disponibile nelle scorte o è esaurito. Nel caso in cui fosse esaurito lancia un’eccezione. Per alcuni prodotti verifica anche la scadenza con le funzioni descritte alla pag. precedente
gestione_venditeKG	comp_carrello_kg	Calcola il prezzo totale dei prodotti (misurati in kg) che vogliono essere venduti e i punti che si guadagnano tramite l’acquisto. Il prezzo e i punti vengono direttamente

		passati alla classe “Carrello”. Verifica se dopo l’aggiunta al carrello quel le scorte di quel prodotto sono esaurite
cancella_carrelloN	comp_carrello_n	Rimpingua il numero di scorte quando un carrello viene cancellato e la vendita non viene effettuata. Pone l’attributo esaurito a NULL
check_venditaN	comp_carrello_n	Verifica se il numero di prodotti che vogliono essere venduti sono disponibili nelle scorte o sono esauriti. Nel caso in cui fossero esauriti lancia un’eccezione. Per alcuni prodotti verifica anche la scadenza con le funzioni descritte alla pag. precedente
gestione_venditeN	comp_carrello_n	Calcola il prezzo totale dei prodotti (per pezzo) che vogliono essere venduti e i punti che si guadagnano tramite l’acquisto. Il prezzo e i punti vengono direttamente passati alla classe “Carrello”. Verifica se dopo l’aggiunta al carrello quel le scorte di quel prodotto sono esaurite
gestione_tessere	fattura	Aggiorna i punti della tessera del cliente quando viene acquistato il prodotto e emessa la fattura