

# **PROGETTO BASI DI DATI**

## **Simulazione applicazione di food delivery**

---



## **1.Progettazione concettuale**

### **1.1 Requisiti iniziali**

Si deve progettare la base di dati per Cibora, un servizio di food delivery per gestire i dati dei ristoranti, degli utenti con i loro relativi ordini e dei fattorini che effettuano le consegne.

Ogni utente necessita di registrarsi con un nome, email, password, numero di telefono e indirizzo di recapito, successivamente, l'utente deve inserire un metodo di pagamento e ricaricare il proprio borsellino. Il borsellino ha un saldo che viene aggiornato ad ogni ordinazione e l'utente può ricaricare il proprio borsellino in qualsiasi momento. Inoltre, gli utenti possono aderire ad una modalità premium che garantisce una priorità sugli ordini.

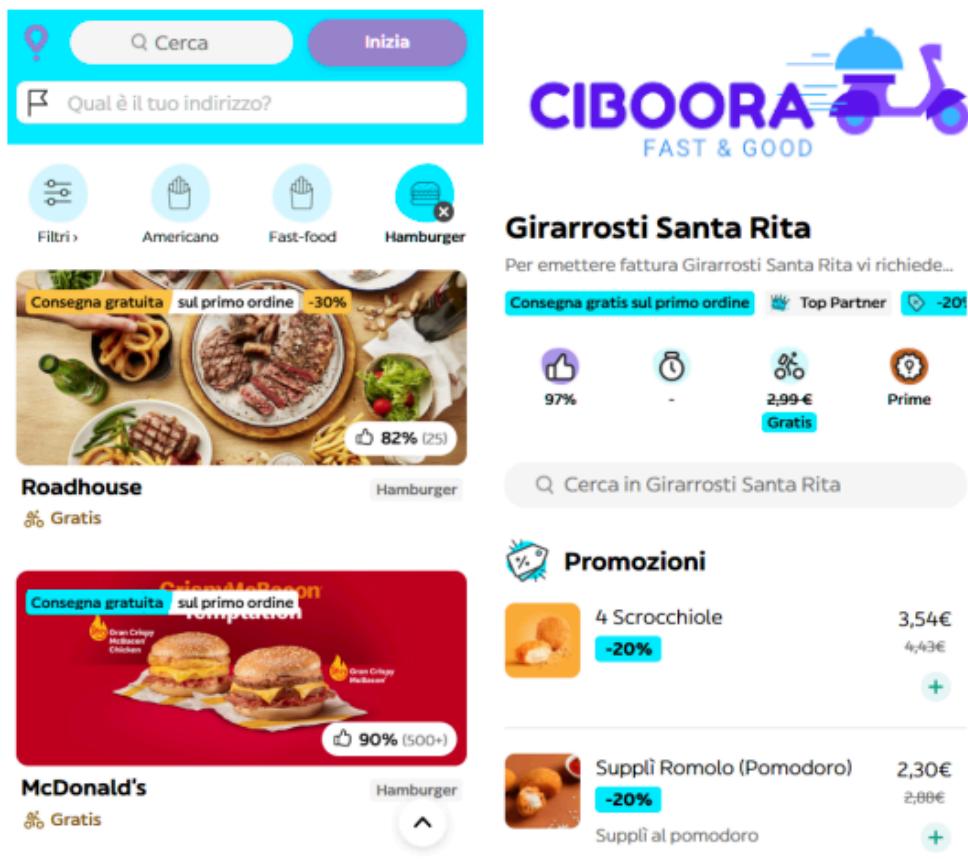
L'utente può accumulare codici sconto in base al numero di ordini effettuati in passato.

I ristoranti sono rappresentati da: nome, descrizione, indirizzo, costo della spedizione, immagine del profilo e un numero di stellette aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al tipo di cibo offerto.

I ristoranti che dimostrano di saper garantire un ottimo servizio (almeno 20 ordini consegnati correttamente, una valutazione clienti maggiore o uguale a 4.5 stelline su cinque, una percentuale massima di ordini annullati dal ristorante dell'1.5%, una percentuale massima di ordini con reclami del 2.5%) sono considerati Top Partner. I Top Partner compaiono in sezioni dedicate all'interno dell'app mobile Cibora e ricevono uno speciale badge che attesta il loro servizio eccellente, aiutando ad aumentare la credibilità e ottenere la fiducia dei clienti. Per i Top Partner si vuole tenere traccia della data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti una lista di piatti da ordinare. Ogni portata ha un titolo, un'immagine, una lista di ingredienti, una lista di allergeni, il prezzo e un eventuale sconto. Inoltre, ogni piatto appartiene ad una o più liste (es. i più venduti, promozioni, dolci, salato, ecc.).

Ogni utente può selezionare una lista di pietanze ed effettuare l'ordine. Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dai clienti, sia dai ristoratori. Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio al ristorante.



Il sistema gestisce un numero arbitrario di riders dove ogni rider è identificato da un codice, dallo stato, dalla posizione aggiornata in tempo reale tramite GPS. I riders sono classificati in base al tipo di mezzo che utilizzano. I riders che utilizzano il monopattino devono indicare quanti km possono effettuare prima che si scarichi la batteria.

Al momento dell'ordine, il sistema trova il rider libero con la somma minima della distanza dal ristorante più la distanza dall'utente. Tuttavia, per ordini che prevedano un tragitto "*posizione corrente del rider-> ristorante-> cliente*" superiore ai 10 km, solo i rider con bici elettrica vengono interpellati. Per monitorare le prestazioni dei ciclofattorini, si vuole tenere traccia del numero di consegne effettuate da ognuno, del momento in cui il cibo da consegnare viene affidato ad un rider e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato al cliente.

Dopo che l'ordine è stato effettuato l'utente ha la possibilità di chattare sia con il ristorante che con il rider in caso ci fossero dei problemi con l'ordine come mancata consegna o netto ritardo.

Quando l'ordine è consegnato l'utente può recensire il ristorante e il rider con una valutazione da 1 a 5 e un commento testuale. Il commento testuale è facoltativo. Inoltre è anche presente la possibilità di dare una mancia al rider per la consegna.

Una volta al mese, vengono aggiornate le seguenti classifiche:

- Riders più veloci nel consegnare gli ordini
- Cibi più popolari
- Ristoranti con più recensioni positive
- Clienti che hanno speso di più

## 1.2 Glossario termini

Termine	Descrizione	Sinonimi	Collegamenti
utente	Cliente che usufruisce del servizio compiendo ordini.	Cliente	Ristorante, Ordine, Recensione, Borsellino
Ristorante	Attività commerciale di ristorazione che usufruisce del servizio.	Locale	Ordine, Recensione, Piatto
Rider	Fattorino delle consegne dotato di mezzo di trasporto	Ciclofattorino, Fattorino, Corriere	Recensione, Ordine
Ordine	Ordinazione del Cliente al ristorante che verrà consegnata dal rider	Consegna, Ordinazione	utente, Messaggi
Borsellino	Portafogli digitale che rappresenta il saldo in euro di ogni cliente	Wallet, Borsellino elettronico, Borsellino digitale	utente, Codice Sconto,
Piatto	Pietanza/e offerta/e dal ristorante che compone l'ordinazione	Portata, Pietanza	Ristorante, Ordine
Ingredienti	Lista degli ingredienti che compongono un piatto		Piatto
Allergeni	Lista degli allergeni		Piatto

Termine	Descrizione	Sinonimi	Collegamenti
Lista	Liste promozionali per le pietanze		Piatto
Recensione	Espressione di giudizio del cliente nei confronti del rider o del ristorante	Valutazione	utente,Rider
Messaggio	Mezzo di comunicazione tra utente e rider o ristorante	chat	Ordine
Reclamo	lamentela di un cliente relativa ad un ordine	lamentela	Ordine
metodo di pagamento	metodo di pagamento scelto dal cliente per ricaricare il proprio borsellino elettronico	mezzo di pagamento	borsellino
Codice Sconto	sconto a favore di un utente maturato per numero di ordini pregressi	Coupon, sconto	borsellino

## 1.3 Requisiti iniziali rivisitati

Segue testo di requisiti iniziali, a seguire la rivisitazione, le entità sono evidenziate seguendo una associazione con quelle sui requisiti rielaborati.

### Testo Iniziale

Si deve progettare la base di dati per Cibora, un servizio di food delivery per gestire i dati dei ristoranti, degli utenti con i loro relativi ordini e dei fattorini che effettuano le consegne.

Ogni utente necessita di registrarsi con un nome, email, password, numero di telefono e indirizzo di recapito, successivamente, l'utente deve inserire un metodo di pagamento e ricaricare il proprio borsellino. Il borsellino ha un saldo che viene aggiornato ad ogni ordinazione e l'utente può ricaricare il proprio borsellino in qualsiasi momento. Inoltre, gli utenti possono aderire ad una modalità premium che garantisce una priorità sugli ordini.

L'utente può accumulare codici sconto in base al numero di ordini effettuati in passato.

I ristoranti sono rappresentati da: nome, descrizione, indirizzo, costo della spedizione, immagine del profilo e un numero di stellette aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al tipo di cibo offerto.

I ristoranti che dimostrano di saper garantire un ottimo servizio (almeno 20 ordini consegnati correttamente, una valutazione clienti maggiore o uguale a 4.5 stelline su cinque, una percentuale massima di ordini annullati dal ristorante dell'1.5%, una percentuale massima di ordini con reclami del 2.5%) sono considerati Top Partner. I Top Partner compaiono in sezioni dedicate all'interno dell'app mobile Cibora e ricevono uno speciale badge che attesta il loro servizio eccellente, aiutando ad aumentare la credibilità e ottenere la fiducia dei clienti. Per i Top Partner si vuole tenere traccia della data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti una lista di piatti da ordinare. Ogni portata ha un titolo, un'immagine, una lista di ingredienti, una lista di allergeni, il prezzo e un eventuale sconto. Inoltre, ogni piatto appartiene ad una o più liste (es. i più venduti, promozioni, dolci, salato, ecc.).

Ogni utente può selezionare una lista di pietanze ed effettuare l'ordine. Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dai clienti, sia dai ristoratori. Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio al ristorante.



 Qual è il tuo indirizzo?

- Filtri >
  Americano
  Fast-food
  Hamburger



**Roadhouse** Hamburger

 Gratis



**McDonald's** Hamburger

 Gratis



## Girarrosti Santa Rita

Per emettere fattura Girarrosti Santa Rita vi richiede...

Consegna gratis sul primo ordine  Top Partner  -20%

 97%

 -

 2,99€  
Gratis

 Prime

## Promozioni


 4 Scrocchiole 3,54€  
 -20% 4,43€


 Supplì Romolo (Pomodoro) 2,30€  
 -20% 2,88€

Il sistema gestisce un numero arbitrario di **riders** dove ogni **rider** è identificato da un codice, dallo stato, dalla posizione aggiornata in tempo reale tramite GPS. I **riders** sono classificati in base al tipo di **mezzo** che utilizzano. I **riders** che utilizzano il monopattino devono indicare quanti km possono effettuare prima che si scarichi la batteria.

Al momento dell'**ordine**, il sistema trova il **rider** libero con la somma minima della distanza dal **ristorante** più la distanza dall'**utente**. Per **ordini** che prevedano un tragitto "*posizione corrente del rider -> ristorante -> cliente*" superiore ai 10 km, solo i **riders** con bici elettrica vengono interpellati. Per monitorare le prestazioni dei **ciclofattorini**, si vuole tenere traccia del numero di consegne effettuate da ognuno, del momento in cui il cibo da consegnare viene affidato ad un **rider** e, per le consegne già completate, anche dell'ora in cui l'**ordine** è stato recapitato al **cliente**.

Dopo che l'**ordine** è stato effettuato l'**utente** ha la possibilità di **chattare** sia con il **ristorante** che con il **rider** in caso ci fossero dei problemi con l'**ordine** come mancata consegna o netto ritardo.

Quando l'ordine è consegnato l'utente può recensire il ristorante e il rider con una valutazione da 1 a 5 e un commento testuale facoltativo.

Inoltre l'utente può dare una mancia al rider per la consegna.

Una volta al mese, vengono aggiornate le seguenti classifiche:

- Riders più veloci nel consegnare gli ordini utente
- Cibi più popolari
- Ristoranti con più recensioni positive
- Clienti che hanno speso di più

## Testo rivisitato

### Utenti

Gli utenti devono registrarsi fornendo nome, email, password, numero di telefono e indirizzo di recapito. Possono aggiungere un metodo di pagamento e ricaricare il proprio borsellino, il cui saldo viene aggiornato ad ogni ordinazione. Il borsellino può essere ricaricato in qualsiasi momento. Gli utenti possono aderire a una modalità premium che garantisce priorità sugli ordini e accumulare codici sconto in base al numero di ordini effettuati in passato.

### Ristoranti

Ogni ristorante è descritto con nome, descrizione, indirizzo, costo della spedizione, immagine del profilo e un numero di stellette, aggiornato ogni lunedì in base alla percentuale di recensioni positive della settimana precedente. I ristoranti appartengono a una o più categorie in base al tipo di cibo offerto. I ristoranti che soddisfano criteri di qualità, come almeno 20 ordini corretti, una valutazione clienti di almeno 4.5 stelle su 5, una percentuale massima di ordini annullati dell'1.5% e una percentuale massima di ordini con reclami del 2.5%, sono classificati come Top Partner. Per i Top Partner, si tiene traccia della data in cui sono entrati in questa categoria. I ristoranti offrono una serie di piatti, ciascuno con titolo, immagine, lista di ingredienti, lista di allergeni, prezzo ed eventuale sconto. Ogni piatto può appartenere a una o più liste, come i più venduti, promozioni, dolci, salato, ecc.

### Ordini



Gli utenti possono selezionare i piatti desiderati e effettuare un ordine, che può essere annullato finché non è affidato a un rider per la consegna. Nel profilo dell'utente è possibile visualizzare gli ordini passati e inviare reclami ai ristoranti. Dopo l'ordine, gli utenti possono scambiare messaggi sia con il ristorante sia con il rider per risolvere eventuali problemi. Una volta consegnato l'ordine, l'utente può recensire ristoranti e rider con una recensione da 1 a 5 e un commento facoltativo, e dare una mancia al rider.

## Riders

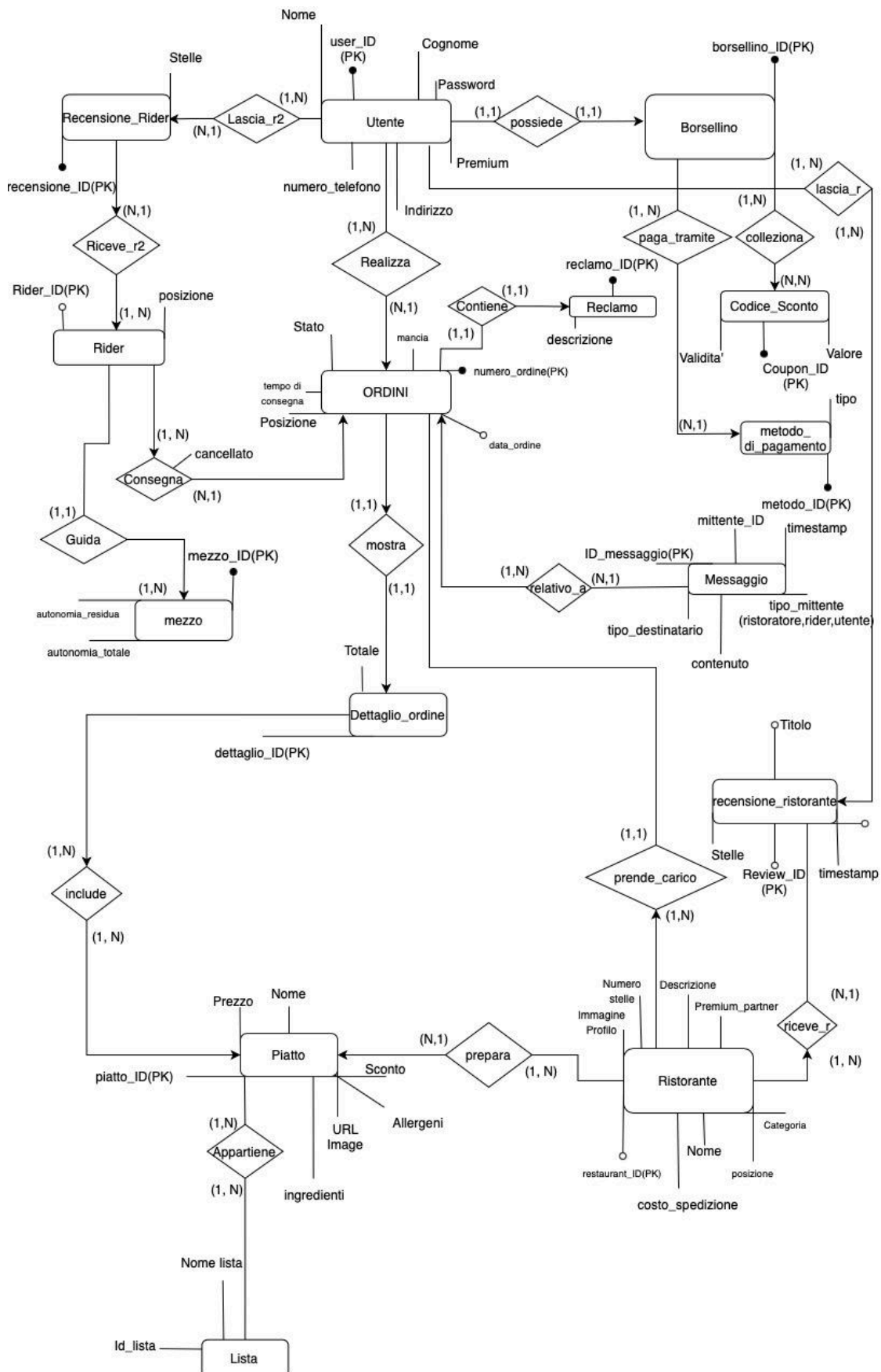
Il sistema gestisce un numero arbitrario di riders, identificati da un codice, stato e posizione aggiornata in tempo reale tramite GPS. I riders sono classificati in base al tipo di mezzo utilizzato. Quelli che usano monopattini devono indicare quanti km possono effettuare prima che si scarichi la batteria. Al momento dell'ordine, il sistema trova il rider libero con la somma minima della distanza dal ristorante e dall'utente. Per tragitti superiori a 10 km, solo i riders con bici elettrica vengono interpellati. Si monitora il numero di consegne effettuate, il momento in cui il cibo viene affidato al rider e l'ora di consegna al cliente.

## Classifiche Mensili

Una volta al mese vengono aggiornate le seguenti classifiche:

- Riders più veloci nel consegnare gli ordini
- Piatti più popolari
- ristoranti con più recensioni positive
- Utenti che hanno speso di più

## 1.4 Modello ER + Business Rules



## Dizionario Dati (*Entità*)

Entità	Descrizione	Attributi	Identificatori
utente	Cliente che usufruisce del servizio compiendo ordini.	user_ID, Nome, Cognome, Email, password, Numero_telefono, Indirizzo, Premium	User_ID
Ristorante	Attività commerciale di ristorazione che usufruisce del servizio.	Nome_ristorante, restaurant_ID, posizione, costo_spedizione, categoria, descrizione, immagine_profilo, premium_partner, numero_stelle	restaurant_ID
Recensione_ristorante	recensione lasciata dall'utente nei confronti del lavoro svolto dal ristorante che ha preparato il suo ordine	review_ID, stelle, restaurant_ID, user_ID, timestamp, titolo	review_ID
Rider	Fattorino delle consegne dotato di mezzo di trasporto	rider_ID, mezzo_ID, posizione	rider_ID
Recensione_rider	recensione lasciata dall'utente nei confronti del lavoro svolto dal rider	recensione_ID, user_ID, rider_ID, commento, stelle	recensione_ID
mezzo	Mezzo di trasporto del rider per consegnare ordini	mezzo_ID, autonomia_residua, rider_ID, autonomia_totale	mezzo_ID
Piatto	pietanza preparata dal locale che andrà a comporre l'ordine	piatto_ID, nome, restaurant_ID, prezzo, dettaglio_ID, sconto immagine, ingredienti,	piatto_ID

Entità	Descrizione	Attributi	Identificatori
		allergeni	
Lista	Lista dei piatti	lista_ID, nome_lista	lista_ID
Ordine	Ordinazione del Cliente al ristorante che verrà consegnata dal rider	user_ID, restaurant_ID, numero_ordine, posizione, data_ordine, rider_ID, tempo_consegna, stato, mancia	numero_ordine
Messaggio	Mezzo di comunicazione tra utente e rider o ristorante	messaggio_ID, mittente_ID, destinatario_ID, timestamp, tipo_mittente, tipo_destinatario, contenuto	messaggio_ID
Dettaglio_ordine	raccolta di dettagli sull'ordine	numero_ordine, dettaglio_ID, totale, ID_piatto, restaurant_ID	dettaglio_ID
reclamo	lamentela del cliente relativa al servizio	reclamo_ID, descrizione	reclamo_ID
Borsellino	Portafogli digitale che rappresenta il saldo in euro di ogni cliente	user_ID, borsellino_ID, saldo	borsellino_ID
metodo_di_pagamento	metodo di pagamento utilizzato dal cliente per la ricarica del borsellino	user_ID, metodo_ID, tipo	metodo_ID
codice_sconto	codici sconto accumulati dall'utente	coupon_ID, validità, Valore	coupon_ID

### Dizionario Dati (*Relazioni*)

Relazione	Descrizione	Componenti	attributi
Possiede	possessione del borsellino elettronico	utente(1,1), borsellino(1,1)	
Realizza	realizzazione dell'ordine da parte dell'utente	Ordine(N,1) utente(1,N)	
lascia_r	realizzazione di recensione del servizio dall'utente nei confronti del ristorante	utente(1,N) recensione_ristorante(1,N)	
Lascia_r2	realizzazione di recensione del servizio dall'utente nei confronti del rider	utente(1,N), recensione_rider(N,1)	
colleziona	relazione di collezione codici sconto nel borsellino	Borsellino(1,N), codice_sconto(N,N)	
paga_tramite	metodo di pagamento scelto per ricarica saldo borsellino	Borsellino(1,N), metodo_di_pagamento(N,1)	
contiene	eventualità per l'ordine di contenere un reclamo	Ordine(1,1), reclamo(1,1)	
mostra	mostra la raccolta dei dettagli specifici legati a un ordine	Ordine(1,1), dettaglio_ordine(1,1)	
relativo_a	mostra correlazione tra un messaggio e un	Ordine(1,N) messaggio(N,1)	

Relazione	Descrizione	Componenti	attributi
	ordine		
prende_carico	presa in carico di un ordine da parte del ristorante	Ordine(1,1), Ristorante(1,N)	
Prepara	il ristorante prepara la/e pietanza/e per l'ordine	Ristorante(1,N), piatto(N,1)	
Riceve_r	ristorante riceve recensione da parte del cliente	Ristorante(1,N); recensione_ristorante(N,1)	
Riceve_r2	rider riceve recensione da parte del cliente	Rider(1,N), recensione_rider(N,1)	
Consegna	consegna dell'ordine da parte del rider	Rider(1,N), ordine(N,1)	cancellato
Guida	relazione di mezzo di trasporto guidato dal rider	Rider(1,1), mezzo(1,N)	
Include	piatto incluso tra i dettagli dell'ordine	Piatto(1,N), Dettaglio_ordine(1,N)	
Appartiene	relazione di appartenenza di un piatto a una lista promozionale	Piatto(1,N), Lista(1,N)	

## Assertzioni

1. **Aggiornamento saldo:**
  - Il saldo deve essere aggiornato ad ogni ordinazione.
2. **Clienti premium:**
  - I clienti iscritti alla modalità premium devono potersi avvalere della possibilità di avere priorità circa la consegna dei loro ordini.
3. **Sconti basati sul numero di ordini:**
  - Il numero di ordini effettuati dai clienti deve influenzare la quantità di sconti che essi riceveranno.
4. **Percorso superiore a 10 km:**
  - Quando gli ordini prevedono un tragitto "posizione del rider > ristorante > cliente" superiore a 10 km, devono essere interpellati solo i rider con bici elettrica.
5. **Recensioni dei ristoranti:**
  - Ogni volta che un utente valuta un ristorante, il numero di stelle del ristorante deve cambiare, ricalcolando il valore ottenuto dalla media aritmetica delle stelle comprendendo anche quelle ricevute con la nuova recensione.
6. **Mancia ai rider:**
  - Nel caso l'utente non sia intenzionato a versare una mancia nei confronti dei rider, il valore di essa deve essere sempre impostato pari a 0 (valore di default).
7. **Mezzi tipo monopattino:**
  - Solo i mezzi di tipo 'Monopattino' devono obbligatoriamente avere un valore per il campo Km.

## 2.1 Tavola dei Volumi

Concetto	Tipo	Volume stimato	Descrizione
utente	Entità	1.000.000	Numero totale di utenti registrati sulla piattaforma
Ristorante	Entità	20.000	Numero totale di ristoranti che utilizzano la piattaforma per la consegna.
Recensione_ristorante	Entità	500.000	Numero totale di recensioni lasciate per i ristoranti, ipotizzando che ogni utente lasci in media 0.5 recensioni.
Rider	Entità	50.000	Numero totale di rider registrati sulla piattaforma.
Recensione_rider	Entità	250.000	Numero totale di recensioni lasciate per i rider, ipotizzando che ogni utente lasci in media 0.25 recensioni per il rider.
Mezzo	Entità	50.000	ogni rider ha un mezzo di trasporto.



Concetto	Tipo	Volume stimato	Descrizione
Piatto	Entità	250.000	Numero totale di piatti disponibili sui vari ristoranti.
Lista	Entità	2.000	Numero totale di liste di piatti (es. offerte speciali, menu di stagione, etc.).
Ordine	Entità	10.000.000	Numero totale di ordini effettuati sulla piattaforma.
Messaggio	Entità	5,000,000	Numero totale di messaggi scambiati tra utenti, rider e ristoranti.
Dettaglio_ordine	Entità	10.000.000	
reclamo	Entità	500,000	Numero totale di reclami ricevuti, ipotizzando che il 5% degli ordini riceva un reclamo.
Borsellino	Entità	1,000,000	Ogni utente ha un borsellino digitale.
metodo_di_pagamento	Entità	1,500,000	Numero totale di metodi di pagamento registrati, ipotizzando che ogni utente abbia 1.5 metodi di pagamento in media.

Concetto	Tipo	Volume stimato	Descrizione
codice_sconto	Entità	3,000,000	Numero totale di codici sconto accumulati dagli utenti, ipotizzando una media di 3 codici sconto per utente.
Possiede	Relazione	1.000.000	Ogni utente possiede un borsellino
Realizza	Relazione	10.000.000	Ogni ordine è realizzato da un utente.
lascia_r	Relazione	500,000	Ogni recensione_r è lasciata da un utente per un ristorante.
Lascia_r2	Relazione	250,000	Ogni recensione_r2 è lasciata da un utente per un rider.
colleziona	Relazione	3,000,000	Ogni borsellino colleziona codici sconto.

Concetto	Tipo	Volume stimato	Descrizione
paga_tramite	Relazione	1,500,000	Ogni borsellino è ricaricato tramite vari metodi di pagamento.
contiene	Relazione	500,000	Ogni reclamo è contenuto in un ordine.
mostra	Relazione	12,000,000	Ogni ordine mostra dettagli dell'ordine.
relativo_a	Relazione	5,000,000	Ogni messaggio è relativo a un ordine.
prende_carico	Relazione	10,000,000	Ogni ordine è preso in carico da un ristorante.
Prepara	Relazione	200,000	Ogni piatto è preparato da un ristorante.
Riceve_r	Relazione	500,000	Ogni recensione_r è ricevuta da un ristorante.
Riceve_r2	Relazione	250,000	Ogni recensione_r2 è ricevuta da un rider.

Concetto	Tipo	Volume stimato	Descrizione
Consegna	Relazione	10,000,000	Ogni ordine è consegnato da un rider.
Guida	Relazione	50,000	Ogni mezzo è guidato da un rider.
Include	Relazione	12,000,000	Ogni dettaglio d'ordine include un piatto.
Appartiene	Relazione	200,000	Ogni piatto appartiene a una lista.

## 2.2 Tavola delle operazioni

Operazione	Tipo	Frequenza
Aggiornamento classifica rider più veloci	Batch	1/mese
Aggiorna classifica cibi più popolari	Batch	1 / mese
Aggiorna classifica ristoranti con più recensioni positive	Batch	1 / mese
Aggiorna classifica clienti che hanno speso di più	Batch	1 / mese
Aggiorna posizione rider in servizio	Batch	1 / 10 secondi
Aggiorna lista Top Partner	Batch	1 / giorno
Utente chatta con ristorante o rider	Interattiva	1 / minuto
Rider con monopattino indica numero chilometri di autonomia	Interattiva	2 / giorno
Calcolo distanza rider - ristorante	Batch	1 / ordine di utente
Dare mancia al rider	Interattiva	1 / ordine di utente

### Note

- I dati per le frequenze sono stime basate sull'attività prevista nei primi tre anni.
- Alcune operazioni, come la posizione dei rider, sono aggiornate frequentemente per garantire l'efficienza del servizio.

## 2.3 Ristrutturazione schema E-R

### 2.3.1 Analisi delle Ridondanze

Ridondanze riscontrate:

- alcune associazioni quali lascia\_r, riceve\_r, lascia\_r2, riceve\_r2 potrebbero essere semplificate usando un modello relazionale più pulito che evita la duplicazione delle informazioni
- un modello con una singola tabella di recensioni piuttosto che due è preferibile per motivi di manutenzione, scalabilità e consistenza dei dati.
- Come attributi, allergeni e ingredienti potrebbero essere ripetuti in più record di piatti.  
Se "Allergeni" e "Ingredienti" sono entità separate, ogni allergene e ingrediente viene memorizzato una sola volta, riducendo la ridondanza dei dati.

Analizzeremo la ridondanza sulla entità delle recensioni:

**Tabella degli Accessi e Numero di Accessi: Modello con Una Singola Tabella di Recensioni**

Operazione	Tipo	Numero di accessi
Inserimento di una recensione rider	Scrittura	1
Inserimento di una recensione ristorante	Scrittura	1
Lettura di tutte le recensioni di un rider	Lettura	1
Lettura di tutte le recensioni di un ristorante	Lettura	1

### **Tabella degli Accessi e Numero di Accessi: Modello con Due Tabelle di Recensioni Distinte**

Operazione	Tipo	Numero di accessi
Inserimento di una recensione rider	Scrittura	1
Inserimento di una recensione ristorante	Scrittura	1
Lettura di tutte le recensioni di un rider	Lettura	1
Lettura di tutte le recensioni di un ristorante	Lettura	1

Come si può vedere a livello di accessi totali per modello sono uguali, ma nonostante ciò, un modello con una singola tabella di recensioni è preferibile per motivi di manutenzione, scalabilità e consistenza dei dati. Nonostante i tipi di accesso simili, la gestione di un'unica tabella semplifica il lavoro di sviluppo e manutenzione, riducendo la ridondanza logica e migliorando la coerenza del database.

### **2.3.2. Eliminazione delle generalizzazioni**

Non sono state riscontrate generalizzazioni da poterne effettuare l'eliminazione.

### **2.3.3. Eventuale partizionamento/accorpamento di entità e associazioni**

La ridondanza analizzata nel paragrafo 2.3.1 è stata risolta con un accorpamento di due entità, ossia recensioni\_rider e recensione\_ristorante sono state accorpate in Recensioni, le associazioni correlate sono state rimosse e aggiunta una sola associazione relativa all'ordine effettuato dal quale è possibile prendere tutti gli attributi necessari.

Tale scelta è preferibile e giustificata a livello di gestione e manutenzione più efficiente nel database nel lungo termine ed assicura coerenza:

- **Manutenzione del Database**
  - **Aggiunta di Nuovi Attributi:** Basta aggiungere il nuovo attributo una sola volta nella tabella recensioni invece di doverne modificare due.
  - **Modifiche Schema:** Le modifiche vengono applicate a una sola tabella, riducendo il rischio di errori.
- **Scalabilità**
  - **Numero di Tabelle:** Meno tabelle rendono il database più facile da gestire e scalare, invece di più al momento di una eventuale espansione del database potrebbero complicare l'organizzazione.
  - **Indici Unificati:** Gli indici vengono creati una sola volta, ottimizzando le operazioni di lettura e scrittura, più indici aumentano il carico di gestione.
- **Consistenza dei Dati**
  - **Nessuna Duplicazione Logica:** Una singola tabella elimina la duplicazione di logica e dati.
  - **Controllo Coerenza:** Mantenere la coerenza dei dati è più semplice quando tutte le recensioni sono in un'unica tabella, con due tabelle assicurare la coerenza richiede più logica di controllo.

### 2.3.4. Eventuale eliminazione degli attributi composti e degli attributi multivalore

È stato scelto di trattare “Ingredienti” e “Allergeni”, inizialmente attributi di “Piatto”, come entità per motivi di riduzione della ridondanza, facilità di aggiornamento, integrità referenziale e flessibilità nell'estendere le informazioni specifiche sugli allergeni e sugli ingredienti, che superano leggermente l'aumento del tempo di accesso dovuto ai join aggiuntivi. Questo approccio offre una soluzione più robusta e scalabile per la gestione dei dati.

#### Riduzione della ridondanza:

- Se "Allergeni" e "Ingredienti" sono entità separate, ogni allergene e ingrediente viene memorizzato una sola volta, riducendo la ridondanza dei dati. Invece, come attributi, gli stessi allergeni e ingredienti potrebbero essere ripetuti in più record di piatti.



### **Facilità di aggiornamento:**

- Aggiornare informazioni sugli allergeni o sugli ingredienti diventa più semplice. Se, ad esempio, cambia il nome di un allergene, è possibile aggiornare un solo record nella tabella "Allergeni", piuttosto che aggiornare ogni piatto che contiene quell'allergene.

### **Possibilità di aggiungere attributi specifici:**

- Allergeni e ingredienti come entità permettono di aggiungere attributi specifici per ognuno di essi. Ad esempio, per gli allergeni, potremmo aggiungere informazioni sulle reazioni comuni, mentre per gli ingredienti potremmo includere informazioni sul valore nutrizionale.

### **Migliore gestione delle relazioni molti-a-molti:**

- Molti piatti possono contenere molti ingredienti e allergeni. Gestire queste relazioni come entità separate permette di utilizzare tabelle associative per modellare correttamente queste relazioni molti-a-molti.

### **Efficienza delle query:**

- Le query che richiedono informazioni dettagliate sugli allergeni o sugli ingredienti possono essere più efficienti. È possibile recuperare tutte le informazioni pertinenti da tabelle separate senza dover duplicare dati in ogni record di piatto.

## **2.3.5. Eventuale scelta degli identificatori principali**

### **Identificatori principali (PK) e loro scelte:**

#### **1. Utente (Email)**

- Email è necessario per distinguere chiaramente ogni utente e gestire le relazioni con altre entità come Recensioni, ORDINI, Messaggio, e Borsellino.

#### **2. Borsellino (borsellino\_ID, Email)**

- borsellino\_ID ed Email identificano in modo univoco ogni borsellino correlato all'Email, facilitando le transazioni e la gestione del saldo.

**3. Codice\_Sconto (coupon\_ID)**

- Coupon\_ID identifica univocamente ogni codice sconto, necessario per applicare correttamente gli sconti agli ordini e tracciare l'utilizzo dei coupon.

**4. Metodo\_di\_Pagamento (metodo\_ID)**

- metodo\_ID identifica univocamente ciascun metodo di pagamento disponibile nel sistema, importante per gestire e distinguere i vari metodi di pagamento utilizzati dagli utenti.

**5. Messaggio (messaggio\_ID, numero\_ordine)**

- messaggio\_ID e numero\_ordine sono una chiave composta che serve a identificare univocamente ogni messaggio relativo ad un ordine inviato o ricevuto nel sistema, essenziale per la gestione delle comunicazioni tra utenti, rider e ristoratori.

**6. Reclamo (reclamo\_ID, numero\_ordine)**

- reclamo\_ID e numero\_ordine identificano univocamente ogni reclamo correlato all'ordine, permettendo di tracciare e gestire efficacemente le segnalazioni e i problemi riscontrati dagli utenti.

**7. Ordini (numero\_ordine)**

- numero\_ordine è un identificatore univoco per ciascun ordine effettuato nel sistema, fondamentale per tracciare gli ordini, associarli a utenti, rider e ristoranti, e gestire lo stato degli ordini.

**8. Dettaglio\_ordine (dettaglio\_ID)**

- dettaglio\_ID è una chiave che identifica univocamente ogni voce di dettaglio all'interno di un ordine, permettendo di gestire e tracciare ogni singolo piatto o articolo ordinato.

**9. Rider (rider\_ID)**

- Rider\_ID è un identificatore univoco per ciascun rider, necessario per gestire le assegnazioni di ordini, le recensioni dei rider e le loro posizioni e stati di consegna.

**10. mezzo (identificatore\_veicolo, rider\_ID)**

- Identificatore\_veicolo identifica univocamente ciascun mezzo di trasporto utilizzato dai rider, importante per la gestione della flotta di veicoli e per tracciare l'autonomia residua e totale dei mezzi.

**11. Recensioni (recensione\_ID)**

- recensione\_ID è un identificatore univoco per ogni recensione nel sistema, consentendo di tracciare recensioni specifiche per rider e ristoranti, collegandole agli utenti che le hanno lasciate.

**12. Ristorante (restaurant\_ID)**

- restaurant\_ID identifica univocamente ciascun ristorante nel sistema, essenziale per gestire i dati dei ristoranti, i menu, le recensioni e gli ordini associati.

### **13. Piatto (piatto\_ID, restaurant\_ID)**

- piatto\_ID e restaurant\_ID compongono un identificatore univoco per ciascun piatto offerto da un ristorante, permettendo di gestire i menu, i dettagli dei piatti, i prezzi e le promozioni.

### **14. Allergeni (Nome)**

- Nome è un identificatore per gli allergeni poiché il nome di un allergene è unico e descrittivo.

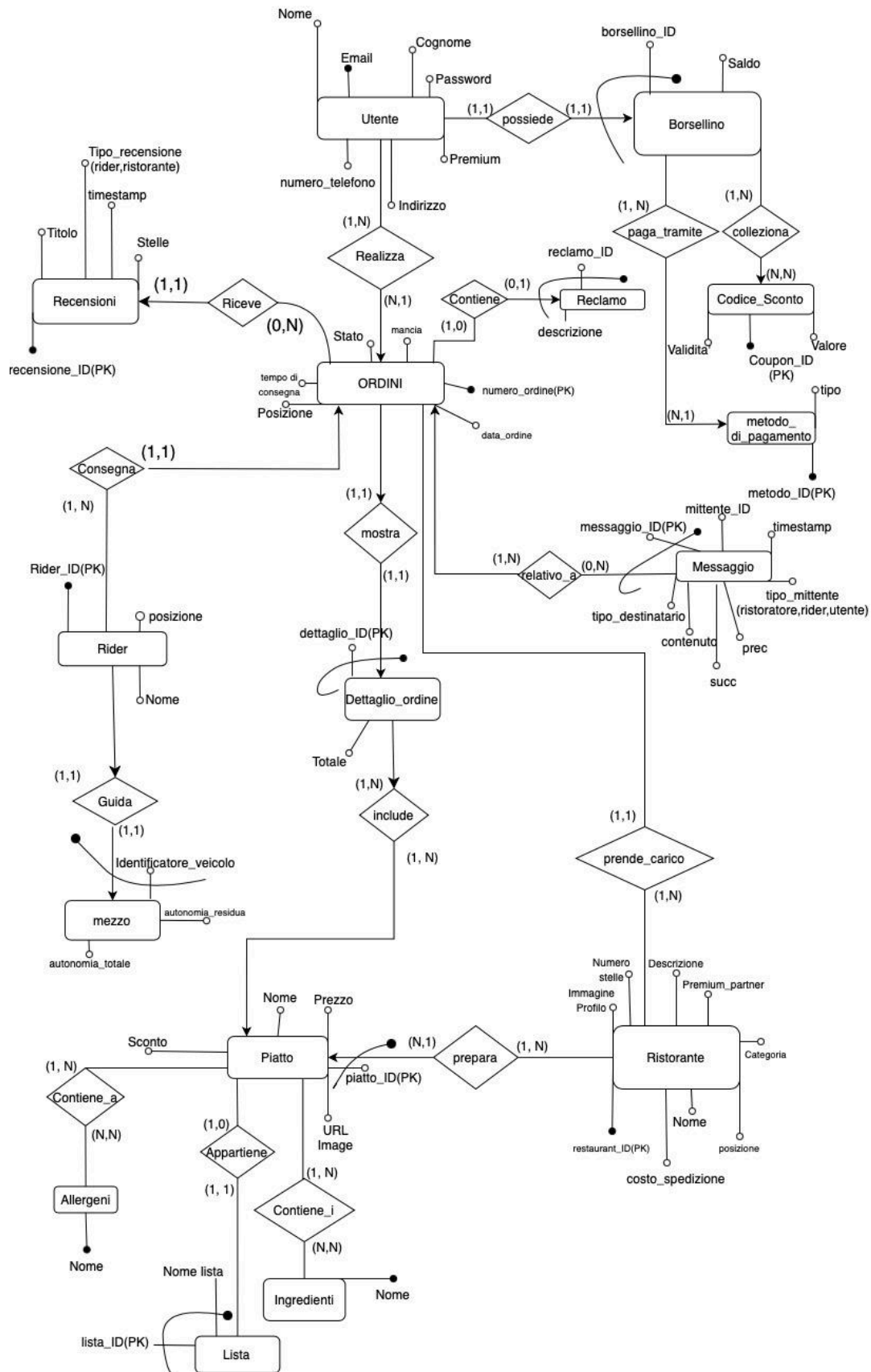
### **15. Ingredienti (Nome)**

- Nome è un identificatore per gli ingredienti poiché i nomi degli ingredienti sono generalmente univoci e sufficientemente descrittivi.

### **16. Lista (lista\_ID, nome\_lista)**

- lista\_ID e nome\_lista compongono un identificatore univoco per ciascuna lista, permettendo di gestire diverse liste di ingredienti o piatti (es. menu speciali, liste di allergeni).

## 28



### Dizionario Dati (*Entità*)

Entità	Descrizione	Attributi	Identificatori
Utente	Cliente che usufruisce del servizio compiendo ordini.	Nome, Cognome, Email, password, Numero_telefono, Indirizzo, Premium	Email
Ristorante	Attività commerciale di ristorazione che usufruisce del servizio.	Nome, restaurant_ID, posizione, costo_spedizione, categoria, descrizione, immagine_profilo, premium_partner, numero_stelle	restaurant_ID
Rider	Fattorino delle consegne dotato di mezzo di trasporto	rider_ID, posizione,nome	rider_ID
mezzo	Mezzo di trasporto del rider per consegnare ordini	Identificatore_veicolo, autonomia_residua, rider_ID, autonomia_totale	mezzo_ID, rider_ID
Piatto	pietanza preparata dal locale che andrà a comporre l'ordine	piatto_ID,nome, restaurant_ID, prezzo, dettaglio_ID, sconto, immagine, ingredienti, allergeni	piatto_ID, restaurant_ID
Lista	Lista dei piatti	lista_ID, nome_lista	lista_ID
Ingredienti	Lista di ingredienti che compongono i piatti	Nome	Nome
Allergeni	Lista di allergeni contenuti nei piatti	Nome	Nome
Ordine	Ordinazione del	user_ID, restaurant_ID,	numero_ordin

Entità	Descrizione	Attributi	Identificatori
	Cliente al ristorante che verrà consegnata dal rider	numero_ordine, posizione, data_ordine, rider_ID, tempo_consegna, stato, mancia	e
Recensioni	Valutazione sul servizio ricevuto da parte del cliente	recensione_ID, Stelle, Titolo, tipo_recensione, timestamp	recensione_ID
Messaggio	Mezzo di comunicazione tra utente e rider o ristorante	messaggio_ID, mittente_ID, destinatario_ID, timestamp, tipo_mittente, tipo_destinatario, contenuto, prec, succ	messaggio_ID
Dettaglio_ordine	raccolta di dettagli sull'ordine	numero_ordine, dettaglio_ID, totale, piatto_ID	numero_ordine, dettaglio_ID
reclamo	lamentela del cliente relativa al servizio	reclamo_ID, descrizione	reclamo_ID
Borsellino	Portafogli digitale che rappresenta il saldo in euro di ogni cliente	user_ID, borsellino_ID, saldo	borsellino_ID
metodo_di_pagamento	metodo di pagamento utilizzato dal cliente per la ricarica del borsellino	user_ID, metodo_ID, tipo	user_ID, metodo_ID
codice_sconto	codici sconto accumulati dall'utente	coupon_ID, validità, Valore	coupon_ID

#### Dizionario Dati (*Relazioni*)

Relazione	Descrizione	Componenti
Possiede	possesso del borsellino elettronico	utente(1,1), borsellino(1,1)
Realizza	realizzazione dell'ordine da parte dell'utente	Ordine(N,1), utente(1,N)
colleziona	relazione di collezione codici sconto nel borsellino	Borsellino(1,N), codice_sconto(N,N)
paga_tramite	metodo di pagamento scelto per ricarica saldo borsellino	Borsellino(1,N), metodo_di_pagamento(N,1)
contiene	eventualità per l'ordine di contenere un reclamo	Ordine(1,0), reclamo(0,1)
mostra	mostra la raccolta dei dettagli specifici legati a un ordine	Ordine(1,1), dettaglio_ordine(1,1)
riceve	correlazione tra ordine e la recensione relativa ad esso	Ordine(0,N), Recensioni(N,1)
relativo_a	mostra correlazione tra un messaggio e un ordine	Ordine(1,N) messaggio(0,N)
prende_carico	presa in carico di un ordine da parte del ristorante	Ordine(1,1), Ristorante(1,N)
Prepara	il ristorante prepara la/e pietanza/e per l'ordine	Ristorante(1,N), piatto(N,1)
Consegna	consegna dell'ordine da parte del rider	Rider(1,N), ordine(N,1)
Guida	relazione di mezzo di trasporto guidato dal rider	Rider(1,1), mezzo(1,1)
Include	piatto incluso tra i dettagli dell'ordine	Piatto(1,N), Dettaglio_ordine(1,N)
Contiene_i	associazione tra piatto e ingredienti da cui è composto	Piatti(1,N), Ingredienti(N,N)
Contiene_a	associazione tra piatto ed	Piatti(1,N),

Relazione	Descrizione	Componenti
	eventuali allergeni inclusi in esso	Allergeni(N,N)
Appartiene	relazione di appartenenza di un piatto a una lista promozionale	Piatto(1,0), Lista(1,1)



## 2.5 Schema relazionale

**Utenti**(Email (pk), nome, cognome, password, numero\_telefono, indirizzo, premium)

**Borsellini**(Email (fk), borsellino\_ID (pk), saldo)

**Metodi\_di\_Pagamenti**(metodo\_ID (pk), tipo, email (fk), borsellino\_ID (fk))

**Codici\_Sconto**(coupon\_id (pk), scadenza, valore)

**Tipi\_di\_Pagamento**(tipo (pk))

**Ordini**(numero\_ordine (pk), posizione, data\_ordine, tempo\_consegna, stato, mancia, restaurant\_id (fk), email (fk), rider\_id (fk), recensione\_id (fk))

**Dettagli\_Ordini**(dettaglio\_id (pk), numero\_ordine (fk), totale, piatto\_id (fk), restaurant\_id (fk))

**Tipo\_Utente**(tipo (pk))

**Messaggi**(messaggio\_id (pk), ordine\_id (fk), mittente\_id (fk), destinatario\_id (fk), timestamp, tipo\_mittente (fk), tipo\_destinatario (fk), contenuto, prec (fk), succ (fk))

**Reclami**(reclamo\_id, descrizione, numero\_ordine (fk))

**Recensioni**(recensione\_id (pk), stelle, titolo, descrizione, tipo\_recensione (fk), timestamp, numero\_ordine (fk))

**Riders**(rider\_ID (pk), mezzo\_ID (fk), nome, posizione)

**Mezzi**(identificatore\_veicolo (pk), autonomia\_residua, rider\_ID (fk), autonomia\_totale)

**Tipo\_Veicolo**(tipo)

**Ristoranti**(restaurant\_ID (pk), nome\_ristorante, posizione, costo\_spedizione, categoria, descrizione, immagine\_profilo, premium\_partner, numero\_stelle)

**Piatti**(piatto\_id, restaurant\_id (fk), nome, descrizione, prezzo, dettaglio\_id (fk))

**Ingredienti**(nome (pk))

**Allergeni**(nome (pk))

**Lista**(nome\_lista, lista\_id)

**Piatto\_Ingredienti**(piatto\_id (fk), restaurant\_id (fk), nome\_ingrediente (fk))

**Piatto\_Allergeni**(piatto\_id (fk), restaurant\_id (fk), nome\_allergene (fk))

**Piatto\_Lista**(piatto\_id (fk), restaurant\_id (fk), nome\_lista, lista\_id)

# 3. Implementazione

## 3.1. DDL di creazione del database.

```
CREATE TABLE
    public.allergeni (nome character varying(50) NOT NULL);
```

```
ALTER TABLE
    public.allergeni
ADD
    CONSTRAINT allergeni_pkey PRIMARY KEY (nome);
```

```
CREATE TABLE
    public.borsellini (
        email character varying(50) NOT NULL,
        borsellino_id integer NOT NULL,
        saldo numeric(10, 2) NOT NULL
    );
```

```
ALTER TABLE
    public.borsellini
ADD
    CONSTRAINT borsellini_pkey PRIMARY KEY (borsellino_id);
```

```
CREATE TABLE
    public.codici_sconto (
        coupon_id character varying(10) NOT NULL,
        scadenza date NULL,
        valore numeric(10, 2) NULL
    );
```

```
ALTER TABLE
    public.codici_sconto
ADD
    CONSTRAINT codici_sconto_pkey PRIMARY KEY (coupon_id);
```

```
CREATE TABLE
    public.dettagli_ordini (
```

```

        dettaglio_id serial NOT NULL,
        numero_ordine bigint NOT NULL,
        totale numeric(10, 2) NULL,
        piatto_id integer NULL,
        restaurant_id integer NULL
    );

ALTER TABLE
    public.dettagli_ordini
ADD
    CONSTRAINT dettagli_ordini_pkey PRIMARY KEY (dettaglio_id);

CREATE TABLE
    public.ingredienti (nome character varying(50) NOT NULL);

ALTER TABLE
    public.ingredienti
ADD
    CONSTRAINT ingredienti_pkey PRIMARY KEY (nome);

CREATE TABLE
    public.lista (
        nome_lista character varying(255) NOT NULL,
        lista_id serial NOT NULL
    );

ALTER TABLE
    public.lista
ADD
    CONSTRAINT lista_pkey PRIMARY KEY (lista_id);

CREATE TABLE
    public.messaggi (
        messaggio_id serial NOT NULL,
        ordine_id bigint NOT NULL,
        mittente_id integer NOT NULL,
        destinatario_id integer NOT NULL,
        "timestamp" timestamp without time zone NOT NULL DEFAULT
CURRENT_TIMESTAMP,
        tipo_mittente character varying(20) NOT NULL,
        tipo_destinatario character varying(20) NOT NULL,
        contenuto text NULL,
        prec serial NOT NULL,

```

```

        succ serial NOT NULL
    );

ALTER TABLE
    public.messaggi
ADD
    CONSTRAINT messaggi_pkey PRIMARY KEY (messaggio_id);

CREATE TABLE
    public.metodi_di_pagamenti (
        metodo_id serial NOT NULL,
        tipo character varying(50) NULL,
        email character varying(50) NULL,
        borsellino_id integer NULL
    );

ALTER TABLE
    public.metodi_di_pagamenti
ADD
    CONSTRAINT metodi_di_pagamenti_pkey PRIMARY KEY (metodo_id);

CREATE TABLE
    public.mezzi (
        identificatore_veicolo character varying(20) NOT NULL,
        autonomia_residua numeric(10, 2) NULL,
        autonomia_totale numeric(10, 2) NULL,
        tipo_veicolo character varying(20) NULL
    );

ALTER TABLE
    public.mezzi
ADD
    CONSTRAINT mezzi_pkey PRIMARY KEY (identificatore_veicolo);

CREATE TABLE
    public.ordini (
        numero_ordine serial NOT NULL,
        posizione text NULL,
        data_ordine timestamp without time zone NOT NULL DEFAULT
CURRENT_TIMESTAMP,
        tempo_consegna integer NULL,
        stato character varying(50) NULL,
        mancia numeric(10, 2) NULL,
        restaurant_id integer NULL,

```

```

        email character varying(50) NOT NULL,
        rider_id serial NOT NULL,
        dettaglio_id serial NOT NULL
    );

ALTER TABLE
    public.ordini
ADD
    CONSTRAINT ordini_pkey PRIMARY KEY (numero_ordine);

CREATE TABLE
    public.piatti (
        piatto_id serial NOT NULL,
        restaurant_id integer NULL,
        nome character varying(100) NULL,
        descrizione text NULL,
        prezzo numeric(10, 2) NULL
    );

ALTER TABLE
    public.piatti
ADD
    CONSTRAINT piatti_pkey PRIMARY KEY (piatto_id);

CREATE TABLE
    public.piatto_allergeni (
        piatto_id integer NOT NULL,
        restaurant_id integer NOT NULL,
        nome_allergene character varying(50) NOT NULL
    );

ALTER TABLE
    public.piatto_allergeni
ADD
    CONSTRAINT piatto_allergeni_pkey PRIMARY KEY
(nome_allergene);

CREATE TABLE
    public.piatto_ingredienti (
        piatto_id integer NOT NULL,
        restaurant_id integer NOT NULL,
        nome_ingrediente character varying(50) NOT NULL
    );

```

```

ALTER TABLE
    public.piatto_ingredienti
ADD
    CONSTRAINT piatto_ingredienti_pkey PRIMARY KEY
(nome_ingrediente);

CREATE TABLE
    public.piatto_lista (
        piatto_id integer NOT NULL,
        restaurant_id integer NOT NULL,
        nome_lista character varying(255) NOT NULL,
        lista_id serial NOT NULL
    );

ALTER TABLE
    public.piatto_lista
ADD
    CONSTRAINT piatto_lista_pkey PRIMARY KEY (lista_id);

CREATE TABLE
    public.reclami (
        reclamo_id serial NOT NULL,
        descrizione text NULL,
        numero_ordine bigint NOT NULL
    );

ALTER TABLE
    public.reclami
ADD
    CONSTRAINT reclami_pkey PRIMARY KEY (numero_ordine);

CREATE TABLE
    public.riders (
        rider_id serial NOT NULL,
        mezzo_id character varying(20) NULL,
        nome character varying(50) NULL,
        posizione text NULL
    );

ALTER TABLE
    public.riders
ADD
    CONSTRAINT riders_pkey PRIMARY KEY (rider_id);

```

```

CREATE TABLE
    public.ristoranti (
        restaurant_id serial NOT NULL,
        nome_ristorante character varying(255) NOT NULL,
        posizione text NULL,
        costo_spedizione numeric(10, 2) NULL,
        categoria character varying(255) NULL,
        descrizione text NULL,
        immagine_profilo text NULL,
        premium_partner boolean NULL DEFAULT false,
        numero_stelle numeric(2, 1) NULL
    );

ALTER TABLE
    public.ristoranti
ADD
    CONSTRAINT ristoranti_pkey PRIMARY KEY (restaurant_id);

CREATE TABLE
    public.tipi_di_pagamento (tipo character varying(50) NOT
NULL);

ALTER TABLE
    public.tipi_di_pagamento
ADD
    CONSTRAINT tipi_di_pagamento_pkey PRIMARY KEY (tipo);

CREATE TABLE
    public.tipo_utente (tipo character varying(20) NOT NULL);

ALTER TABLE
    public.tipo_utente
ADD
    CONSTRAINT tipo_utente_pkey PRIMARY KEY (tipo);

CREATE TABLE
    public.tipo_veicolo (tipo character varying(20) NOT NULL);

ALTER TABLE
    public.tipo_veicolo
ADD
    CONSTRAINT tipo_veicolo_pkey PRIMARY KEY (tipo);

CREATE TABLE

```



```

public.utenti (
    email character varying(50) NOT NULL,
    nome character varying(50) NOT NULL,
    cognome character varying(50) NOT NULL,
    password character varying(30) NOT NULL,
    numero_telefono character varying(30) NULL,
    indirizzo text NULL,
    premium boolean NULL DEFAULT false,
    borsellino_id integer NOT NULL
);

ALTER TABLE
    public.utenti
ADD
    CONSTRAINT utenti_pkey PRIMARY KEY (email);
CREATE TABLE
    public.utilizzo_codici_sconto (
        utilizzo_id serial NOT NULL,
        coupon_id character varying(10) NULL,
        email character varying(50) NULL,
        borsellino_id integer NULL
    );

ALTER TABLE
    public.utilizzo_codici_sconto
ADD
    CONSTRAINT utilizzo_codici_sconto_pkey PRIMARY KEY
    (utilizzo_id);

```

## 3.2. DML di popolamento di tutte le tabelle del database.

```
insert into "allergeni" ("nome") values ('glutine');
insert into "allergeni" ("nome") values ('crostacei');
insert into "allergeni" ("nome") values ('uova');
insert into "allergeni" ("nome") values ('pesce');
insert into "allergeni" ("nome") values ('arachidi');
insert into "allergeni" ("nome") values ('soia');
insert into "allergeni" ("nome") values ('latte');
insert into "allergeni" ("nome") values ('frutta a guscio');
insert into "allergeni" ("nome") values ('sedano');
insert into "allergeni" ("nome") values ('senape');
insert into "allergeni" ("nome") values ('sesamo');
insert into "allergeni" ("nome") values ('anidride
solforosa');
insert into "allergeni" ("nome") values ('lupini');
insert into "allergeni" ("nome") values ('molluschi');

insert into "borsellini" ("borsellino_id", "email", "saldo")
values (1, 'user1@example.com', '0.00');
insert into "borsellini" ("borsellino_id", "email", "saldo")
values (2, 'user2@example.com', '0.00');
insert into "borsellini" ("borsellino_id", "email", "saldo")
values (3, 'user3@example.com', '0.00');
insert into "borsellini" ("borsellino_id", "email", "saldo")
values (4, 'user4@example.com', '0.00');
insert into "borsellini" ("borsellino_id", "email", "saldo")
values (5, 'user5@example.com', '0.00');

insert into "codici_sconto" ("coupon_id", "scadenza",
"valore") values ('ABC123', '2024-12-31', '10.00');
insert into "codici_sconto" ("coupon_id", "scadenza",
"valore") values ('DEF456', '2024-11-30', '5.00');
insert into "codici_sconto" ("coupon_id", "scadenza",
"valore") values ('GHI789', '2024-10-15', '15.00');
insert into "codici_sconto" ("coupon_id", "scadenza",
"valore") values ('JKL012', '2024-09-30', '20.00');
insert into "codici_sconto" ("coupon_id", "scadenza",
"valore") values ('MNO345', '2024-08-31', '8.50');
```

```

insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (21, '21', 1, 1, '10.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (22, '22', 2, 2, '8.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (23, '23', 3, 3, '18.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (24, '24', 4, 4, '7.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (25, '25', 5, 5, '9.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (26, '26', 6, 1, '15.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (27, '27', 7, 2, '10.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (28, '28', 8, 3, '12.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (29, '29', 9, 4, '8.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (30, '30', 10, 5, '11.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (31, '31', 11, 3, '4.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (32, '32', 12, 2, '9.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (33, '33', 13, 3, '8.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (34, '34', 14, 4, '7.00');

```

```

insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (35, '35', 15, 5, '6.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (36, '36', 2, 2, '8.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (37, '37', 1, 1, '7.00');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (38, '38', 4, 3, '8.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (39, '39', 5, 5, '5.50');
insert into "dettagli_ordini" ("dettaglio_id",
"numero_ordine", "piatto_id", "restaurant_id", "totale")
values (40, '40', 3, 4, '7.00');

insert into "ingredienti" ("nome") values ('Pasta');
insert into "ingredienti" ("nome") values ('Uova');
insert into "ingredienti" ("nome") values ('Guanciale');
insert into "ingredienti" ("nome") values ('Pecorino Romano');
insert into "ingredienti" ("nome") values ('Vitello');
insert into "ingredienti" ("nome") values ('Prosciutto
crudo');
insert into "ingredienti" ("nome") values ('Salvia');
insert into "ingredienti" ("nome") values ('Mascarpone');
insert into "ingredienti" ("nome") values ('Cacao');
insert into "ingredienti" ("nome") values ('Mozzarella');
insert into "ingredienti" ("nome") values ('Basilico');
insert into "ingredienti" ("nome") values ('Pomodoro');
insert into "ingredienti" ("nome") values ('Funghi');
insert into "ingredienti" ("nome") values ('Olive');
insert into "ingredienti" ("nome") values ('Riso');
insert into "ingredienti" ("nome") values ('Pesce crudo');
insert into "ingredienti" ("nome") values ('Tofu');
insert into "ingredienti" ("nome") values ('Verdure');
insert into "ingredienti" ("nome") values ('Gamberi');
insert into "ingredienti" ("nome") values ('Frutta e
verdura');
insert into "ingredienti" ("nome") values ('Frutti di bosco');
insert into "ingredienti" ("nome") values ('Pasta frolla');
insert into "ingredienti" ("nome") values ('Cioccolato');

```

```

insert into "ingredienti" ("nome") values ('Mandorle');

insert into "lista" ("lista_id", "nome_lista") values (1,
'Contorni');
insert into "lista" ("lista_id", "nome_lista") values (2,
'Primi piatti');
insert into "lista" ("lista_id", "nome_lista") values (3,
'Secondi piatti');
insert into "lista" ("lista_id", "nome_lista") values (4,
'Pizza');
insert into "lista" ("lista_id", "nome_lista") values (5,
'Sushi');
insert into "lista" ("lista_id", "nome_lista") values (6,
'Insalate');
insert into "lista" ("lista_id", "nome_lista") values (7,
'Dolci');
insert into "lista" ("lista_id", "nome_lista") values (8,
'Bevande');
insert into "lista" ("lista_id", "nome_lista") values (9, 'I
piu venduti');

insert into "messaggi" ("contenuto", "destinatario_id",
"messaggio_id", "mittente_id", "ordine_id", "prec", "succ",
"timestamp", "tipo_destinatario", "tipo_mittente") values
('Buonasera, vorrei confermare la mia ordinazione.', 201, 17,
101, '1', 17, 17, '2024-07-04 18:53:42.540918', 'Ristoratore',
'Utente');
insert into "messaggi" ("contenuto", "destinatario_id",
"messaggio_id", "mittente_id", "ordine_id", "prec", "succ",
"timestamp", "tipo_destinatario", "tipo_mittente") values
('Gentile cliente, grazie per il suo ordine. La preparazione è
in corso.', 101, 18, 201, '1', 18, 18, '2024-07-04
18:53:42.540918', 'Utente', 'Ristoratore');
insert into "messaggi" ("contenuto", "destinatario_id",
"messaggio_id", "mittente_id", "ordine_id", "prec", "succ",
"timestamp", "tipo_destinatario", "tipo_mittente") values
('Buongiorno, ho bisogno di assistenza riguardo al mio
ordine.', 202, 19, 102, '2', 19, 19, '2024-07-04
18:53:42.540918', 'Ristoratore', 'Utente');
insert into "messaggi" ("contenuto", "destinatario_id",
"messaggio_id", "mittente_id", "ordine_id", "prec", "succ",
"timestamp", "tipo_destinatario", "tipo_mittente") values
('Certo, posso aiutarla con qualche informazione?', 102, 20,

```

```

202, '2', 20, 20, '2024-07-04 18:53:42.540918', 'Utente',
'Ristoratore');

insert into "metodi_di_pagamenti" ("borsellino_id", "email",
"metodo_id", "tipo") values (1, 'user1@example.com', 1,
'PayPal');
insert into "metodi_di_pagamenti" ("borsellino_id", "email",
"metodo_id", "tipo") values (2, 'user2@example.com', 2,
'Satispay');
insert into "metodi_di_pagamenti" ("borsellino_id", "email",
"metodo_id", "tipo") values (3, 'user3@example.com', 3, 'Carta
di credito');
insert into "metodi_di_pagamenti" ("borsellino_id", "email",
"metodo_id", "tipo") values (4, 'user4@example.com', 4,
'Bonifico bancario');
insert into "metodi_di_pagamenti" ("borsellino_id", "email",
"metodo_id", "tipo") values (5, 'user5@example.com', 5,
'PayPal');

insert into "mezzi" ("autonomia_residua", "autonomia_totale",
"identificatore_veicolo", "tipo_veicolo") values ('40.00',
'40.00', 'VEH-001', 'Monopattino');
insert into "mezzi" ("autonomia_residua", "autonomia_totale",
"identificatore_veicolo", "tipo_veicolo") values ('100.00',
'100.00', 'VEH-002', 'Bici Elettrica');
insert into "mezzi" ("autonomia_residua", "autonomia_totale",
"identificatore_veicolo", "tipo_veicolo") values ('100.00',
'100.00', 'VEH-003', 'Bici');
insert into "mezzi" ("autonomia_residua", "autonomia_totale",
"identificatore_veicolo", "tipo_veicolo") values ('40.00',
'40.00', 'VEH-004', 'Monopattino');
insert into "mezzi" ("autonomia_residua", "autonomia_totale",
"identificatore_veicolo", "tipo_veicolo") values ('100.00',
'100.00', 'VEH-005', 'Bici Elettrica');

insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 21, 'user1@example.com', '2.00', 21, 'Via
Roma 1', 1, 1, 'Consegnato', 30);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04

```

```

17:22:34.494283', 22, 'user2@example.com', '1.50', 22, 'Via
Garibaldi 2', 2, 2, 'In Consegna', 25);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 23, 'user3@example.com', '0.00', 23, 'Piazza
Duomo 3', 3, 3, 'In Preparazione', 40);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 24, 'user4@example.com', '3.00', 24, 'Corso
Italia 4', 4, 4, 'Consegnato', 20);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 25, 'user5@example.com', '0.00', 25, 'Via
Dante 5', 5, 5, 'Cancellato', 35);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 26, 'user2@example.com', '2.50', 26, 'Via
Roma 1', 1, 1, 'Consegnato', 28);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 27, 'user3@example.com', '1.75', 27, 'Via
Garibaldi 2', 2, 2, 'In Consegna', 22);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 28, 'user4@example.com', '0.00', 28, 'Piazza
Duomo 3', 3, 3, 'In Preparazione', 38);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 29, 'user5@example.com', '2.00', 29, 'Corso
Italia 4', 4, 4, 'Consegnato', 18);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 30, 'user1@example.com', '0.00', 30, 'Via
Dante 5', 5, 5, 'Cancellato', 32);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",

```

```

"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 31, 'user3@example.com', '1.50', 31, 'Via
Roma 1', 1, 1, 'Consegnato', 26);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 32, 'user4@example.com', '1.00', 32, 'Via
Garibaldi 2', 2, 2, 'In Consegna', 24);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 33, 'user5@example.com', '0.00', 33, 'Piazza
Duomo 3', 3, 3, 'In Preparazione', 36);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 34, 'user1@example.com', '2.50', 34, 'Corso
Italia 4', 4, 4, 'Consegnato', 16);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 35, 'user2@example.com', '0.00', 35, 'Via
Dante 5', 5, 5, 'Cancellato', 30);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 36, 'user4@example.com', '3.00', 36, 'Via
Roma 1', 1, 1, 'Consegnato', 34);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 37, 'user5@example.com', '1.25', 37, 'Via
Garibaldi 2', 2, 2, 'In Consegna', 20);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 38, 'user1@example.com', '0.00', 38, 'Piazza
Duomo 3', 3, 3, 'In Preparazione', 42);
insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 39, 'user2@example.com', '2.75', 39, 'Corso
Italia 4', 4, 4, 'Consegnato', 15);

```



```

insert into "ordini" ("data_ordine", "dettaglio_id", "email",
"mancia", "numero_ordine", "posizione", "restaurant_id",
"rider_id", "stato", "tempo_consegna") values ('2024-07-04
17:22:34.494283', 40, 'user3@example.com', '0.00', 40, 'Via
Dante 5', 5, 5, 'Cancellato', 33);

insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Pasta con uova, guanciale
e pecorino romano', 'Spaghetti alla Carbonara', 1, '10.50',
1);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Fettine di vitello con
prosciutto crudo e salvia', 'Saltimbocca alla Romana', 2,
'15.00', 1);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Dolce italiano al caffè
con mascarpone e cacao', 'Tiramisù', 3, '7.00', 1);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Pizza con pomodoro,
mozzarella e basilico', 'Pizza Margherita', 4, '8.00', 2);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Pizza con pomodoro,
mozzarella, prosciutto cotto, funghi e olive', 'Pizza
Capricciosa', 5, '10.00', 2);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Pizza chiusa a forma di
mezzaluna con ripieno di pomodoro e mozzarella', 'Calzone', 6,
'9.50', 2);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Torta al cioccolato e
mandorle tipica della tradizione partenopea', 'Torta Caprese',
7, '6.50', 2);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Assortimento di pesce
crudo tagliato a fettine', 'Sashimi Misto', 8, '18.00', 3);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Sushi con riso e pezzo di
pesce crudo o altro ingrediente sopra', 'Nigiri Sushi', 9,
'4.50', 3);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Frittura giapponese di
gamberi e verdure in pastella croccante', 'Tempura', 10,
'12.00', 3);

```

```

insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Tiramisù al tè verde
matcha giapponese', 'Matcha Tiramisu', 11, '8.50', 3);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Hamburger con carne di
manzo, lattuga, pomodoro e salsa', 'Hamburger Classico', 12,
'7.00', 4);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Hamburger con formaggio
cheddar fuso', 'Cheeseburger', 13, '8.00', 4);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Patatine tagliate a
bastoncino e fritte', 'Patatine Fritte', 14, '3.50', 4);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Insalata mista con verdure
fresche e tofu', 'Insalata Vegana', 15, '9.00', 5);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Lasagne con ragù di
verdure e besciamella', 'Lasagne Vegetariane', 16, '11.50',
5);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Frullato di frutta e
verdura per purificare il corpo', 'Smoothie Detox', 17,
'6.50', 5);
insert into "piatti" ("descrizione", "nome", "piatto_id",
"prezzo", "restaurant_id") values ('Dolce con base di pasta
frolla e frutti di bosco freschi', 'Crostata ai Frutti di
Bosco', 18, '5.50', 5);

insert into "piatto_allergeni" ("nome_allergene", "piatto_id",
"restaurant_id") values ('uova', 1, 1);

insert into "piatto_ingredienti" ("nome_ingrediente",
"piatto_id", "restaurant_id") values ('Uova', 1, 1);
insert into "piatto_ingredienti" ("nome_ingrediente",
"piatto_id", "restaurant_id") values ('Pasta', 1, 1);
insert into "piatto_ingredienti" ("nome_ingrediente",
"piatto_id", "restaurant_id") values ('Guanciale', 1, 1);

insert into "piatto_lista" ("lista_id", "nome_lista",
"piatto_id", "restaurant_id") values (9, 'I piu venduti', 1,
1);

```

```

insert into "riders" ("mezzo_id", "nome", "posizione",
"rider_id") values ('VEH-001', 'John', 'Via Roma 1', 1);
insert into "riders" ("mezzo_id", "nome", "posizione",
"rider_id") values ('VEH-002', 'Emma', 'Via Garibaldi 2', 2);
insert into "riders" ("mezzo_id", "nome", "posizione",
"rider_id") values ('VEH-003', 'Michael', 'Via Dante 3', 3);
insert into "riders" ("mezzo_id", "nome", "posizione",
"rider_id") values ('VEH-004', 'Sophia', 'Via Manzoni 4', 4);
insert into "riders" ("mezzo_id", "nome", "posizione",
"rider_id") values ('VEH-005', 'James', 'Corso Vittorio
Emanuele 5', 5);

insert into "ristoranti" ("categoria", "costo_spedizione",
"descrizione", "immagine_profilo", "nome_ristorante",
"numero_stelle", "posizione", "premium_partner",
"restaurant_id") values ('Italiana', '3.50', 'Ristorante con
cucina tradizionale italiana', 'url_immagine1', 'Gusto
Italiano', '4.5', 'Via Roma 1', true, 1);
insert into "ristoranti" ("categoria", "costo_spedizione",
"descrizione", "immagine_profilo", "nome_ristorante",
"numero_stelle", "posizione", "premium_partner",
"restaurant_id") values ('Pizza', '4.00', 'Specializzato in
pizza napoletana', 'url_immagine2', 'Pizza Napoletana', '4.2',
'Via Garibaldi 2', false, 2);
insert into "ristoranti" ("categoria", "costo_spedizione",
"descrizione", "immagine_profilo", "nome_ristorante",
"numero_stelle", "posizione", "premium_partner",
"restaurant_id") values ('Sushi', '5.00', 'Ristorante
giapponese con sushi fresco', 'url_immagine3', 'Sushi Dreams',
'4.8', 'Piazza Duomo 3', true, 3);
insert into "ristoranti" ("categoria", "costo_spedizione",
"descrizione", "immagine_profilo", "nome_ristorante",
"numero_stelle", "posizione", "premium_partner",
"restaurant_id") values ('Fast Food', '3.00', 'Offre panini e
piatti veloci', 'url_immagine4', 'Burger Station', '3.9',
'Corso Italia 4', false, 4);
insert into "ristoranti" ("categoria", "costo_spedizione",
"descrizione", "immagine_profilo", "nome_ristorante",
"numero_stelle", "posizione", "premium_partner",
"restaurant_id") values ('Vegetariano', '3.20', 'Cucina
vegetariana e vegana', 'url_immagine5', 'Green Garden', '4.6',
'Via Dante 5', true, 5);

insert into "tipi_di_pagamento" ("tipo") values ('PayPal');

```

```

insert into "tipi_di_pagamento" ("tipo") values ('Satispay');
insert into "tipi_di_pagamento" ("tipo") values ('Carta di
credito');
insert into "tipi_di_pagamento" ("tipo") values ('Bonifico
bancario');

insert into "tipo_utente" ("tipo") values ('Utente');
insert into "tipo_utente" ("tipo") values ('Ristoratore');
insert into "tipo_utente" ("tipo") values ('Rider');

insert into "tipo_veicolo" ("tipo") values ('Monopattino');
insert into "tipo_veicolo" ("tipo") values ('Bici');
insert into "tipo_veicolo" ("tipo") values ('Bici Elettrica');

insert into "utenti" ("borsellino_id", "cognome", "email",
"indirizzo", "nome", "numero_telefono", "password", "premium")
values (1, 'Bagala', 'user1@example.com', 'Via Roma 1',
'Samuele', '1234567890', 'password1', true);
insert into "utenti" ("borsellino_id", "cognome", "email",
"indirizzo", "nome", "numero_telefono", "password", "premium")
values (2, 'Barrile', 'user2@example.com', 'Via Garibaldi 2',
'Giuseppe', '0987654321', 'password2', false);
insert into "utenti" ("borsellino_id", "cognome", "email",
"indirizzo", "nome", "numero_telefono", "password", "premium")
values (3, 'Verdi', 'user3@example.com', 'Via Dante 3',
'Giulia', NULL, 'password3', false);
insert into "utenti" ("borsellino_id", "cognome", "email",
"indirizzo", "nome", "numero_telefono", "password", "premium")
values (4, 'Neri', 'user4@example.com', 'Via Manzoni 4',
'Anna', '2345678901', 'password4', true);
insert into "utenti" ("borsellino_id", "cognome", "email",
"indirizzo", "nome", "numero_telefono", "password", "premium")
values (5, 'Gialli', 'user5@example.com', 'Corso Vittorio
Emanuele 5', 'Paolo', '3456789012', 'password5', true);

insert into "utilizzo_codici_sconto" ("borsellino_id",
"coupon_id", "email", "utilizzo_id") values (1, 'ABC123',
'user1@example.com', 1);
insert into "utilizzo_codici_sconto" ("borsellino_id",
"coupon_id", "email", "utilizzo_id") values (2, 'DEF456',
'user2@example.com', 2);
insert into "utilizzo_codici_sconto" ("borsellino_id",
"coupon_id", "email", "utilizzo_id") values (3, 'GHI789',
'user3@example.com', 3);

```

### 3.3. Operazioni di cancellazione e modifica

-----  
Aumenta del 10% il valore del coupon in scadenza oggi

```
UPDATE public.codici_sconto  
SET valore = valore * 1.10  
WHERE scadenza = CURRENT_DATE;
```

-----  
Cancellato tutti gli ordini del ristorante Gusto Italiano

```
DELETE FROM public.ordini  
WHERE restaurant_id = (SELECT restaurant_id FROM  
public.ristoranti WHERE nome_ristorante = 'Gusto Italiano');
```

-----  
Aggiunta ingredienti Tiramisu al ristorante 1

```
insert into "piatto_ingredienti" ("nome_ingrediente",  
"piatto_id", "restaurant_id") values ('Mascarpone', 3, 1);
```

```
insert into "piatto_ingredienti" ("nome_ingrediente",  
"piatto_id", "restaurant_id") values ('Cioccolato', 3, 1);
```