

Simone Pollano (matr: 1068650)
Kevin Roberto (matr: 1085068)

RELAZIONE PROGETTO BASE DI DATI

Indice:

1. Progettazione concettuale:

- 1.1. Requisiti iniziali
- 1.2. Glossario dei termini
- 1.3. Requisiti rivisti
- 1.4. Schema E-R con Business rules

2. Progettazione logica:

- 2.1. Tavola dei volumi
- 2.2. Tavola delle operazioni
- 2.3. Ristrutturazione dello schema E-R:
 - 2.3.1. Analisi delle ridondanze
 - 2.3.2. Eliminazioni delle generalizzazioni
 - 2.3.3. Eventuale scelta degli identificatori principali
- 2.4. Schema E-R ristrutturato con Business rules
- 2.5. Schema relazionale

3. Implementazione:

- 3.1. DDL di creazione del database
- 3.2. DML di popolamento di tutte le tabelle del database
- 3.3. Operazioni di cancellazione e modifica per verificare i vincoli e gli effetti causati dalle operazioni su chiavi esterne.

1. Progettazione concettuale

1.1. Requisiti iniziali

Si deve progettare la base di dati per Cibora (Figura 1(a)), un innovativo servizio di food delivery per gestire i dati dei ristoranti aderenti, degli utenti con i loro relativi ordini e dei fattorini che effettuano le consegne in bicicletta.

Per beneficiare del servizio, ogni utente deve registrarsi inserendo nome, e-mail, password, numero di telefono, indirizzo di recapito.

Una volta registratosi, l'utente deve inserire un mezzo di pagamento (es.: carta di credito, PayPal, Satispay) e ricaricare il proprio borsellino elettronico.

Il borsellino ha un saldo che viene aggiornato ad ogni ordinazione e l'utente

Inoltre, gli utenti possono sottoscrivere la modalità premium che garantisce una priorità sugli ordini. può ricaricare il proprio borsellino in qualsiasi momento.

L'utente può collezionare codici di sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.

Ogni ristorante (Figura 1(b)) è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un'immagine di 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 (ad esempio: fast food, vegetariano, ...).

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 (occupato/disponibile/fuori servizio), dalla posizione aggiornata in tempo reale tramite GPS.

I riders sono classificati in base al tipo di mezzo che utilizzano (bicicletta normale, bicicletta elettrica, monopattino).

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 sia 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

Figura 1:

- (a) La lista dei ristoranti con filtro “Hamburger”.
 (b) I dettagli di un ristorante.

The image shows two screenshots of a mobile application interface for food delivery.

Screenshot (a): List of Restaurants with Filter "Hamburger".

- Top Bar:** Includes a location icon, a search bar with placeholder "Q Cerca" and button "Inizia", and a text input field "Qual è il tuo indirizzo?".
- Filters:** Buttons for "Filtri", "Americano", "Fast-food", and "Hamburger" (highlighted).
- Restaurants:**
 - Roadhouse:** Shows a banner for "Consegna gratuita sul primo ordine -30%" with a photo of steaks and fries. Below it, there's a rating of "82% (25)".
 - McDonald's:** Shows a banner for "Crispy McRoon Temptation" with a photo of three burgers. Below it, there's a rating of "90% (500+)".

Screenshot (b): Details of a Restaurant.

- Header:** Includes a location icon, a search bar with placeholder "Qual è il tuo indirizzo?", and an information icon.
- Restaurant Logo:** Girarrosto Santa Rita logo featuring a bread roll.
- Restaurant Name:** **Girarrosto Santa Rita**
- Description:** Per emettere fattura Girarrosto Santa Rita vi richiede...
- Offer:** Consegnata gratis sul primo ordine, Top Partner, -20%.
- Metrics:** 97% (like), - (clock), 2,99 € (Gratis), Prime.
- Search Bar:** Q Cerca in Girarrosto Santa Rita.
- Promotions:**
 - 4 Scrocchiate:** 3,54 € (from 4,43 €), -20%, + (add).
 - Supplì Romolo (Pomodoro):** 2,30 € (from 2,88 €), -20%, + (add).

1.2. Glossario dei termini

TERMINI	DESCRIZIONE	SINONIMI O OMONIMI	COLLEGAMENTI
Ristorante	Luogo di provenienza del cibo		Utente, Piatto, Rider, Ordine, Chat
Utente	Individuo che ordina il cibo	Cliente	Ristorante, Piatto, Rider, Chat, Borsellino
Rider	Individuo che consegna il cibo	Fattorini, ciclofattorini	Ristorante, Utente, Ordine, Chat
Piatto	Piatto da mangiare	Cibo, pietanze, portate	Ristorante, Utente, Ordine
Ordine	Piatto ordinato sull'app	Ordinazione	Piatto, Rider, Utente
Borsellino	Contenitore del saldo inserito dall'utente		Utente
Stelline	Metodo di valutazione	Stellette, valutazione	Ristorante
Chat	Messaggio inviato da un individuo	Messaggio	Utente, Rider, Ristorante

1.3. Requisiti rivisti

Si deve progettare la base di dati per Cibora (Figura 1(a)), un innovativo servizio di food delivery per gestire i dati dei ristoranti aderenti, degli utenti con i loro relativi ordini e dei rider che effettuano le consegne.

Ogni utente deve registrarsi inserendo nome, e-mail, password, numero di telefono, indirizzo di recapito, un metodo di pagamento e ricaricare il proprio borsellino.

Il borsellino ha un saldo che viene aggiornato ad ogni ordine e può essere ricaricato dall'utente in qualsiasi momento.

Inoltre, gli utenti possono sottoscrivere la modalità premium che garantisce priorità sugli ordini.

L'utente può collezionare codici sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.

Ogni ristorante (Figura 1(b)) è rappresentato da un nome, una descrizione, un indirizzo, il costo di spedizione, un'immagine profilo e un numero di stelline aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana.

Ogni ristorante appartiene ad una o più categorie in base al tipo di cibo offerto (ad esempio: fast food, vegetariano, ...).

I ristoranti che dimostrano di saper garantire un ottimo servizio avendo almeno 20 ordini consegnati correttamente, una valutazione clienti maggiore o uguale a 4.5 / 5 stelline, una percentuale massima di ordini annullati dal ristorante del 1.5% e una percentuale massima di ordine con reclami del 2.5% sono considerati Top Partner.

I Top Partner compaiono in una sezione dedicata all'interno dell'app e ricevono un badge che attesta il loro servizio eccellente.

Per i Top Partner si vuole sapere la data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti una lista di piatti da ordinare.

Ogni piatto ha un titolo, un'immagine, una lista di ingredienti, una lista di allergeni, il prezzo e un eventuale sconto e ha una o più liste di appartenenza (per esempio: i più venduti, promozioni, dolci, salato, ecc.).

Ogni utente può selezionare una lista di piatti ed effettuare l'ordine; finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dai clienti sia dai ristoranti.

Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio sulla chat al ristorante.

Il sistema gestisce un numero arbitrario di rider dove ognuno di questi è identificato da un codice, dallo stato (occupato/disponibile/fuori servizio), dalla posizione GPS.

I riders sono classificati in base al tipo di mezzo che utilizzano (bicicletta normale, bicicletta elettrica, monopattino).

I riders che utilizzano i 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 più la distanza dall'utente.

Tuttavia, per ordini che prevedono un tragitto "posizione corrente del rider → ristorante → utente" superiore ai 10 km, solo i rider con bici elettrica vengono interpellati.

Per monitorare le prestazioni dei riders, si deve sapere il numero di consegne effettuate da ognuno, il momento in cui il cibo da

consegnare viene affidato ad un rider e, per le consegne già completate, anche l'ora in cui l'ordine è stato recapitato al cliente. Dopo che l'ordine è stato completato l'utente ha la possibilità di messaggiare sia con il ristorante sia con il rider in caso ci fossero dei problemi con l'ordine come mancata consegna o netto ritardo. Quando l'ordine è stato consegnato l'utente può recensire il ristorante e il rider con una valutazione da 1 a 5 e un commento testuale (facoltativo). Inoltre, è possibile dare una mancia al rider per la consegna (facoltativo). Una volta al mese, vengono aggiornate le seguenti classifiche:

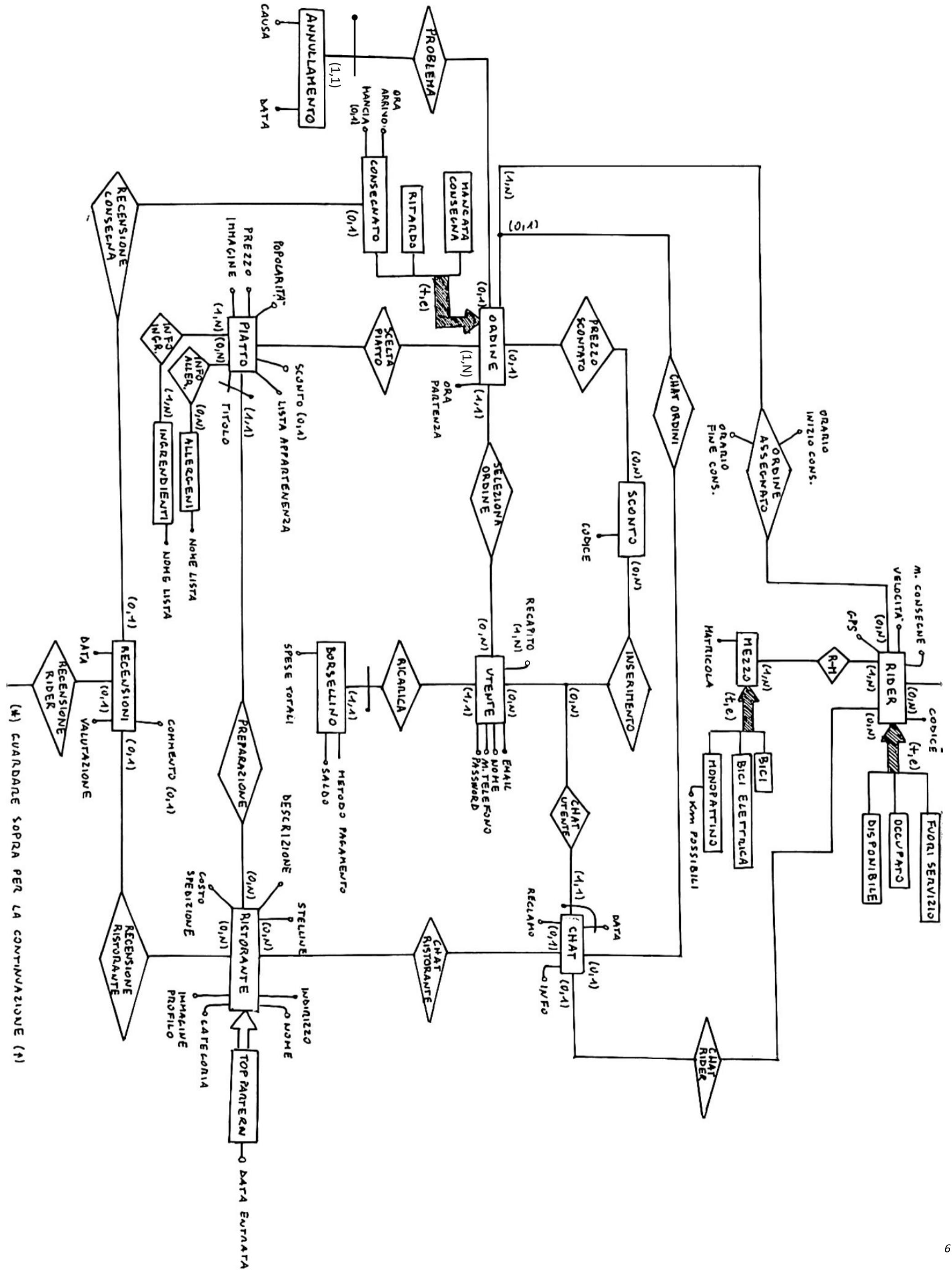
- Riders più veloci nel consegnare gli ordini
- Cibi più popolari
- Ristoranti con più recensioni positive
- Clienti che hanno speso di più

• Frasi di carattere generale	• Frasi relative agli utenti
<p>Si deve progettare la base di dati per un innovativo servizio di food delivery per gestire i dati dei ristoranti aderenti, degli utenti con i loro relativi ordini e dei fattorini che effettuano le consegne.</p>	<p>Ogni utente deve registrarsi inserendo nome, e-mail, password, numero di telefono, indirizzo di recapito, il metodo di pagamento e ricaricare il proprio borsellino.</p> <p>L'utente può collezionare codici sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.</p> <p>Dopo che l'ordine è stato completato l'utente ha la possibilità di messaggiare sia con il ristorante sia con il rider in caso ci fossero dei problemi con l'ordine.</p> <p>Quando l'ordine è stato consegnato l'utente può recensire il ristorante e il rider con una valutazione da 1 a 5 e un commento testuale.</p>

• Frasi relative ai ristoranti	• Frasi relative agli ordini
<p>Ogni ristorante è rappresentato da un nome, una descrizione, un indirizzo, il costo di spedizione, un'immagine profilo e un numero di stelline.</p> <p>Ogni ristorante appartiene ad una o più categorie in base al tipo di cibo offerto.</p> <p>I ristoranti propongono agli utenti una lista di piatti da ordinare.</p> <p>I ristoranti che dimostrano di saper garantire un ottimo servizio sono considerati Top Partner.</p>	<p>L'utente può collezionare codici di sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.</p> <p>Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dai clienti sia dai ristoranti.</p> <p>Nel profilo dell'utente si possono ispezionare gli ordini passati.</p>

• Frasi relative ai riders	• Frasi relative al piatto
<p>Il sistema gestisce un numero arbitrario di rider dove ognuno di questi è identificato da un codice, dallo stato (occupato/disponibile/fuori servizio), dalla posizione GPS.</p> <p>I riders sono classificati in base al tipo di mezzo che utilizzano (bicicletta normale, bicicletta elettrica, monopattino).</p> <p>I riders che utilizzano i monopattini devono indicare quanti Km possono effettuare prima che si scarichi la batteria.</p> <p>Al momento dell'ordine, il sistema trova il rider libero con la somma minima della distanza dal ristorante più la distanza dall'utente.</p>	<p>I ristoranti propongono agli utenti una lista di piatti da ordinare.</p> <p>Ogni piatto ha un titolo, un'immagine, una lista di ingredienti, una lista di allergeni, il prezzo e un eventuale sconto.</p> <p>Inoltre, ogni piatto ha una o più liste di appartenenza (per esempio: i più venduti, promozioni, dolci, salato, ecc.)</p>

1.4. Schema E-R e Business rules



Business rules

- Dizionario dei dati sull'entità:

ENTITA'	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Utente	Individuo che usa l'app Cibora per ordinare il cibo	E-mail, Nome, n° telefono, Password, Recapito (1,N)	E-mail
Borsellino	Borsellino ricaricabile internamente dall'utente contenente un saldo utilizzabile per effettuare pagamenti	Saldo, Metodo pagamento, Spese totali	Utente
Ordine	Ordine di piatti che si vuole effettuare	Ora partenza	Ora partenza
Annullamento	in caso il piatto consegnato abbia delle problematiche si può annullare la consegna specificandone la causa.	Causa, Data	Ordine
Piatto	Piatto proposto dal ristorante e scelto dall'utente	Sconto (0,1), Immagine, Lista appartenenza, Titolo, Prezzo, Popolarità	Titolo, Ristorante
Allergeni	Lista contenente eventuali allergeni presenti nel piatto	Nome lista	Nome lista
Ingredienti	Lista contenente gli ingredienti presenti nel piatto	Nome lista	Nome lista
Ristorante	Luogo dove vengono cucinati i piatti	Descrizione, Costo spedizione, Indirizzo, Nome, Categoria, Immagine profilo, Stelline	Indirizzo
Chat	Chat in cui utenti, ristoranti e riders possono comunicare dove l'utente, per gli ordini passati può effettuare un reclamo	Data, Reclamo, Info	Data, Utente
Rider	Persona che tramite un mezzo consegna il piatto comprato dall'utente	Codice, n° consegne, GPS, Velocità	Codice
Sconto	Rappresenta lo sconto che un utente può utilizzare ed è accumulato in base agli ordini effettuati	Codice	Codice
Recensioni	Sezione in cui recensire la consegna, il rider o il ristorante tramite una valutazione da 1 a 5 e un commento tutti e due facoltativi.	Valutazione, Commento (0,1), Data	Data
Mezzo	Mezzo utilizzato dal rider per consegnare l'ordine	Matricola	Matricola

- Dizionario dei dati sulle generalizzazioni:

Generalizzazione di Ordine:

TIPOLOGIA	DESCRIZIONE	ENTITIA' FIGLIE
Totale, esclusiva	Spiega lo stato dell'ordine	Consegnato (<u>attributi</u> : Mancia(0,1), Ora arrivo), Ritardo, Mancata consegna

Generalizzazione di Ristorante:

TIPOLOGIA	DESCRIZIONE	ENTITÀ FIGLIE
Parziale	In base ad una buona condotta, ad un buon servizio e ad altri fattori i ristoranti possono ottenere il badge top partner per aumentare lo status quo del ristorante nell'app	Top partner (<u>attributo</u> : Data entrata)

Generalizzazione di Rider (stato):

TIPOLOGIA	DESCRIZIONE	ENTITÀ FIGLIE
Totale, esclusiva	Rappresenta lo stato del rider	Fuori servizio, Occupato, Disponibile

Generalizzazione di Rider (mezzo):

TIPOLOGIA	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORI	ENTITÀ FIGLIE
Totale, esclusiva	Rappresenta i diversi mezzi che un rider può utilizzare per trasportare un ordine	Matricola	Matricola	Bici, Bici elettrica, Monopattino (<u>attributo</u> : Km possibili)

- Dizionario dei dati sulle associazioni:

ASSOCIAZIONI	DESCRIZIONE	COMPONENTI	ATTRIBUTI
Selezione ordine	Permette di effettuare un ordine	Utente(0,N) Ordine (1,1)	
Prezzo scontato	Permette <u>applicare</u> lo sconto sull'ordine	Ordine(0,1) Sconto(0,N)	
Inserimento	Permette di <u>inserire</u> uno sconto sull'ordine	Utente(0,N) Sconto(0,N)	
Ricarica	Permette di ricaricare il saldo del borsellino	Utente(1,1) Borsellino(1,1)	
Scelta piatto	Permette di selezionare il piatto che si vuole ordinare	Ordine(1,N) Piatto(0,N)	
Preparazione	Permette di inviare l'ordine al ristorante in modo da poterlo preparare	Piatto(1,1) Ristorante(0,N)	
Info allergeni	Permette di poter leggere la lista degli allergeni	Piatto(0,N) Allergeni(0,N)	
Info ingredienti	Permette di poter leggere la lista degli ingredienti	Piatto(1,N) Ingredienti(1,N)	
Chat utente	Permette all'utente di utilizzare la chat	Utente(0,N) Chat(1,1)	
Chat ristorante	Permette al ristorante di utilizzare la chat	Ristorante(0,N) Chat(0,1)	
Chat rider	Permette al rider di utilizzare la chat	Rider(0,N) Chat(0,1)	
Chat ordini	Permette di scrivere un reclamo	Ordine(0,1) Chat(0,1)	
Ordine assegnato	Permette di assegnare l'ordine al rider	Ordine(1,N) Rider(0,N)	Orario inizio consegna, Orario fine consegna
R-M (tipologia mezzo)	Permette al rider di utilizzare un mezzo per spostarsi	Rider(1,N) Mezzo(1,N)	
Recensione ristorante	Permette di recensire il ristorante	Recensioni(0,1) Ristorante(0,N)	
Recensione rider	Permette di recensire il rider	Recensioni(0,1) Rider(0,N)	
Recensione consegna	Permette di recensire la consegna	Recensioni(0,1) Consegnato (0,1)	

Problema	Permette in caso di un problema dopo la consegna di annullare l'ordine	Ordine(0,1), Annullamento (0,1)	
-----------------	--	---------------------------------	--

- Vincoli di integrità:

REGOLE DI VINCOLO
(1) Lo sconto, per essere utilizzato, non deve essere usato più di una volta dallo stesso utente
(2) Il reclamo, se lo si vuole fare, deve essere effettuato solo su un ordine passato
(3) Un ristorante per essere top partner deve soddisfare diversi requisiti (vedi da riga 16 a riga 19)
(4) Un ordine Se la “posizione corrente del rider → ristorante → utente” è superiore ai 10 km deve essere selezionati solo i rider dotati di bici elettrica
(5) Un utente, se vuole, deve fare reclamo solo su ordini passati

- Derivazioni:

REGOLE DI DERIVAZIONE
Il prezzo totale pagato dall'utente si ottiene... $\text{costo totale} = \text{prezzo} + \text{costo spedizione} - \text{sconto} + \\ - \text{saldo} + \text{mancia}$
Il numero di stelline si ottiene... $\text{stelline} = \frac{1}{n} \sum_{i=1}^n x_i$

2. Progettazione logica

2.1. Tavola dei volumi

Per poter elaborare la tavola dei volumi si ipotizza il possibile flusso di “dati” per ogni entità e per ogni associazione.

- Viene ipotizzato che aderiscano 5.000 utenti
- Viene ipotizzato che aderiscano 100 ristoranti e che ogni ristorante proponga 10 piatti
- Viene ipotizzato che vengano assunti 400 riders in modo tale da poter coprire ogni ristorante con almeno 4 riders
- Il numero di ordini è di circa 500
- il numero di borsellini utilizzabili è pari al numero di utenti, così come il numero di annullamenti
- Ipoteticamente il numero di ingredienti massimo presenti in un piatto è di 5 ingredienti
- Ipoteticamente il numero di allergeni massimo presenti in un piatto è di 3 (glutine, uova, pomodoro)
- Il numero di chat disponibili è al più di 3 siccome tutte e tre le entità collegate alla chat (Utente, Rider, Ristorante) possono comunicare tra loro;
 - Le chat tra utente e rider rispecchiano il numero di ordini effettuati e quindi da consegnare: in media 500 al giorno
 - Le chat tra utente e ristorante si basano sulle informazioni richieste dagli utenti e non dagli ordini: in media 100 al giorno
 - La chat tra ristorante e rider si basa sul numero di consegne da fare, quindi, sul numero di ordini da consegnare: in media 500 al giorno
- Il numero di inserimenti di sconto è in media di 300 al giorno (non in tutti gli ordini viene inserito lo sconto)
- Il numero di ricariche del borsellino in media è 5.000 siccome vi sono ipoteticamente 5.000 utenti attivi
- Il numero massimo di sconti accumulabili è di 15
- Il numero di recensioni dipende dal numero di utenti online, di ordini consegnati, di riders impiegati e di ristoranti in attività, ipoteticamente ~ 1000
- Il numero di selezione ordine disponibili è poco più degli ordini effettuati ed iniziati, circa 550

- Il numero di piatti con prezzo scontato è di un piatto per ogni ristorante quindi 100
- Le interazioni con la chat saranno:
 - Per utente 600
 - Per il ristorante 600
 - Per il rider 1000
- Ipoteticamente per ogni ordine si possono ordinare 3 piatti; quindi, verranno preparati massimo 3 piatti a ordine
- Ipoteticamente il numero di informazioni sugli ingredienti è di 10
- Ipoteticamente il numero di informazioni sugli allergeni è di 3 essendoci 3 tipologie di allergeni
- Le recensioni, invece, essendoci ipoteticamente 1000 recensioni vi saranno ~ 300 recensioni per i ristoranti, ~ 200 per i riders e ~ 500 per le consegne
- Ipoteticamente il numero di problematiche riscontrate è di 50 (un ordine su 10)
- Ipoteticamente il numero di annullamenti è di 25 (uno ogni 20 ordini)
- Essendoci 400 Rider di conseguenza ci saranno anche 400 mezzi
- Il numero di interazioni R-M sono 400 siccome ogni rider deve avere un proprio mezzo
- Ipoteticamente vengono assegnati 150 ordini ai rider

CONCETTO	TIPO	VOLUME
Utente	E	5.000
Borsellino	E	5.000
Ordine	E	500
Annullamento	E	25
Piatto	E	1.000
Allergeni	E	3
Ingredienti	E	500
Ristorante	E	100
Chat	E	1000 (400+100+500)
Rider	E	400
Sconto	E	15
Recensioni	E	1000
Mezzo	E	400
Selezione ordine	A	550
Prezzo scontato	A	100
Inserimento	A	300
Ricarica	A	5.000
Scelta piatto	A	10
Preparazione	A	1000
Info allergeni	A	10
Info ingredienti	A	3
Chat utente	A	1000
Chat ristorante	A	600
Chat rider	A	1000
Recensione ristorante	A	300
Recensione rider	A	200
Recensione consegna	A	500
problematiche	A	50
R-M	A	400
Ordine assegnato	A	150

2.2. Tavola delle operazioni

Per poter elaborare la tavola delle operazioni bisogna lavorare sulle cardinalità delle associazioni tra le varie entità.

- Per operazioni di tipo interattivo (I) quando un'operazione prevede l'interazione con l'utente e viene eseguita in tempo reale
- Per operazioni di tipo batch (B) quando un'operazione viene eseguita automaticamente senza interazione con l'utente

(I numeri dell'elenco puntato corrispondono con i numeri della tabella delle operazioni)

1. L'utente inizia un ordine ~ 40 volte al giorno
 - È di tipo I siccome l'utente interagisce con l'ordine
2. Viene inserito uno sconto su un piatto ~ 18 volte al giorno
 - È di tipo I siccome viene inserito dall'utente
3. Viene aggiunto un piatto nell'ordine ~ 40 volte al giorno; viene ottenuto dal numero di ordini giornalieri siccome ogni ordine deve avere per forza un piatto
 - È di tipo I siccome è l'utente a scegliere il piatto
4. Si ricarica il saldo del borsellino ~ 1 volta a settimana
 - È di tipo I siccome è l'utente ad aggiornare lo stato del borsellino aggiungendo dei soldi
5. Aggiorna il n° di consegne del rider ~ 1 al giorno; viene effettuata a fine giornata
 - È di tipo I siccome è direttamente collegato alle scelte dell'utente ?
6. Viene rilasciata una recensione al ristorante ~ 10 alla settimana
 - È di tipo I siccome viene rilasciata dall'utente
7. Viene rilasciata una recensione al rider ~ 7 alla settimana (in media una al giorno)
 - È di tipo I siccome viene rilasciata dall'utente
8. Viene rilasciata una recensione alla consegna ~ 20 alla settimana
 - È di tipo I siccome viene rilasciata dall'utente
9. Viene memorizzato un nuovo utente ~ 5 volte al giorno
 - È di tipo I siccome prevede un'iterazione diretta con l'utente per registrarsi
10. Viene memorizzato un nuovo ristorante ~ 3 volte alla settimana
 - È di tipo I siccome il Ristorante o per lo meno colui che effettua l'iscrizione è un utente effettivo
11. Viene memorizzato un nuovo rider ~ 5 volte alla settimana
 - È di tipo I siccome il rider è un utente effettivo
12. Si aggiorna lo stato del rider ~ 3 volte al giorno siccome il rider deve fare delle pause
(pausa metà mattina, pausa pranzo, pausa metà pomeriggio) e può essere irreperibile in determinate fasce giornaliere
 - È di tipo I perché il rider è un utente effettivo
13. Viene effettuata la valutazione giornaliera del ristorante
 - È di tipo I siccome è un evento direttamente coinvolto con gli utenti
14. Si aggiorna lo stato dell'ordine ~ 120 volte al giorno
 - È di tipo I siccome lo stato viene aggiornato da una persona esterna: il rider
15. L'ordine viene preso in carica da un ristorante ~ 50 volte al giorno
 - È di tipo I siccome l'ordine è preso da una persona esterna: colui che accetta gli ordini
16. Per ogni ordine si assegna un rider ~ 40 volte al giorno; viene ottenuto dal numero di ordini giornalieri
 - È di tipo B siccome il rider viene assegnato automaticamente

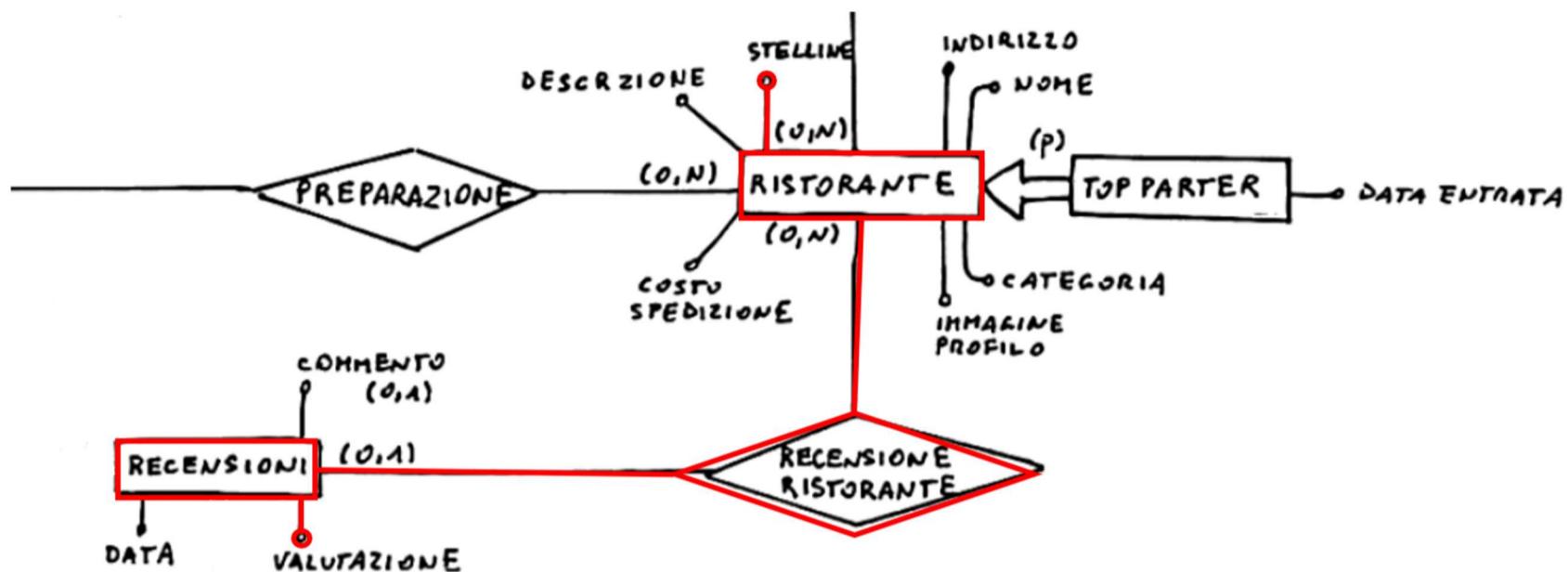
OPERAZIONE	DESCRIZIONE	TIPO	FREQUENZA
1	L'utente inizia l'ordine	I	40 / giorno
2	Inserimento di uno sconto su un piatto	I	18 / giorno
3	Viene aggiunto un piatto nell'ordine	I	40 / giorno
4	Ricarica del saldo nel borsellino	I	1 / settimana
5	Aggiungi il numero di consegne fatte dal rider	I	1 / giorno
6	Viene rilasciata una recensione al ristorante	I	7 / settimana
7	Viene rilasciata una recensione al rider	I	7 / settimana
8	Viene rilasciata una recensione alla consegna	I	20 / settimana
9	Memorizza un nuovo utente	I	5 / giorno
10	Memorizza un nuovo ristorante	I	3 / Settimana
11	Memorizza un nuovo rider	I	5 / settimana
12	Aggiorna lo stato del rider	I	3 / giorno

13	Ottieni tutti i dati del ristorante comprese le stelline	I	1 / giorno
14	Aggiorna lo stato dell'ordine	I	120 / giorno
15	Presa in carico di un ordine da un ristorante	I	50 / giorno
16	L'ordine viene assegnato ad un rider	B	40 / giorno

2.2. Ristrutturazione dello schema E-R

2.3.1. Analisi delle ridondanze

Si è deciso di analizzare la ridondanza individuata nella seguente interazione, nello specifico tra l'attributo *stelline* di *Ristorante* e *valutazione* di *Recensione*.



Preleviamo dalla tavola dei volumi i concetti utili in questa interazione e dalla tavola delle operazioni tutte le operazioni effettuate.

CONCETTO	TIPO	VOLUME
Ristorante	E	100
Recensioni	E	1000
Recensione ristoranti	A	300

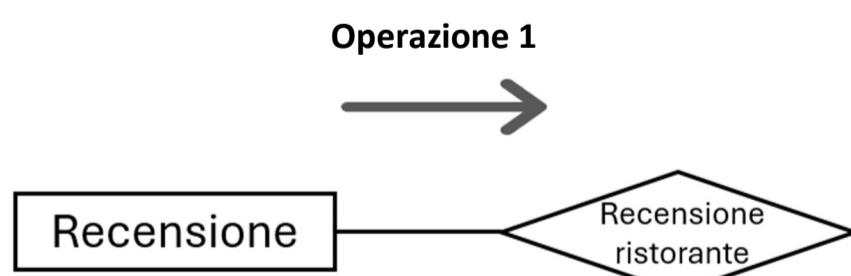
OPERAZIONE	DESCRIZIONE	TIPO	FREQUENZA
6	Viene rilasciata una recensione al ristorante	I	7 / settimana
16	Ottieni tutti i dati del ristorante comprese le stelline	I	5 / giorno

Per poter analizzare al meglio la ridondanza individuata, serve produrre una tavola degli accessi dove tener conto dell'operazione effettuata dall'entità o dalla relazione. Queste operazioni possono essere:

- Operazioni in scrittura: si verifica un inserimento di dati
- Operazioni in lettura: si verifica un'estrapolazione di dati

Siccome più pesanti gli accessi in scrittura si devono considerare due volte.

Scenario A: in assenza di ridondanza



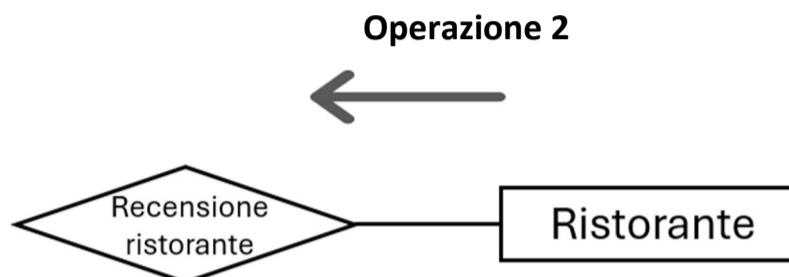
- Tavola degli accessi
 - Per l'entità Recensione si effettua un accesso in scrittura per inserire la recensione
 - Per la relazione Recensione ristorante si effettua un accesso in scrittura per inviare la recensione
- Si verificano quindi due accessi in scrittura.

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Recensione	Entità	1	S
Recensione ristorante	Associazione	1	S

- Questa istruzione viene eseguita ~ 5 volte al giorno

$$TOT\ ACCESSI = 7 * (n^{\circ} accessi\ in\ scrittura * 2) = 7 * (2 * 2) = 28$$

Scenario A: in assenza di ridondanza



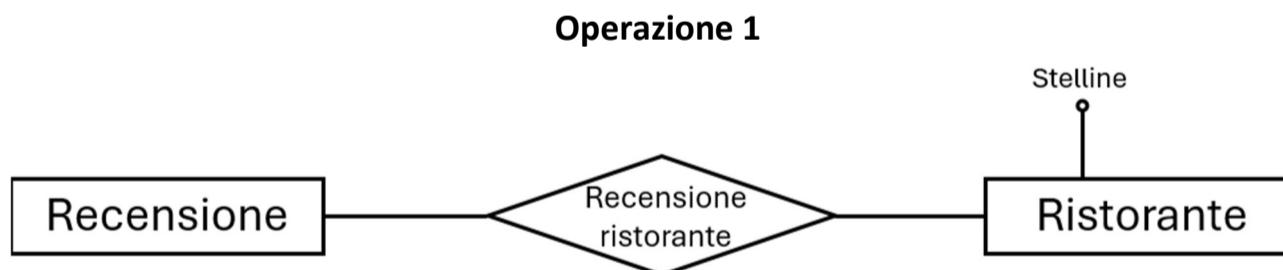
- Tavola degli accessi
 - Per l'entità Ristorante si effettua un accesso in lettura per leggere le stelline del ristorante
 - Per l'associazione Recensione ristorante si effettuano 3 accessi in lettura per ricevere le recensioni
$$n^{\circ} accessi\ in\ lettura = n^{\circ} recensioni\ ristorante / n^{\circ} ristoranti = 300 / 100 = 3$$
- Per l'entità Recensione si effettuano 300 accessi in lettura essendoci 300 recensioni adibite ai ristoranti

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ristorante	Entità	1	L
Recensione ristorante	Associazione	3	L
Recensione	Entità	300	L

- Questa istruzione viene eseguita ~ 5 volte al giorno

$$TOT\ ACCESSI = 5 * n^{\circ} accessi\ in\ lettura = 5 * 304 = 1520$$

Scenario B: in presenza di ridondanza



- Tavola degli accessi
 - Per l'entità Recensioni si effettua un accesso in scrittura per inserire la recensione
 - Per l'associazione Recensione ristorante si effettua un accesso in scrittura per inviare la recensione
 - Per l'entità Ristorante si effettua un accesso in lettura per leggere le stelline e un accesso in scrittura per modificare l'attributo stelline.

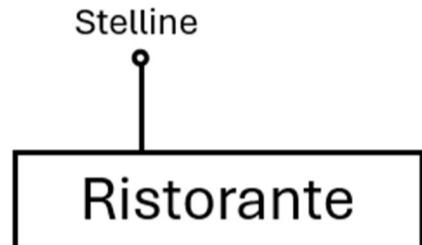
CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ristorante	Entità	1	S
Recensione ristorante	Associazione	1	S
Ristorante	Entità	1	L
Ristorante	Entità	1	S

- Questa istruzione viene eseguita ~ 7 volte al giorno

$$TOT\ ACCESSI = 7 * (n^{\circ} accessi\ in\ scrittura * 2) + 7 * (n^{\circ} accessi\ in\ lettura) = 7 * (3 * 2) + 7 * 1 = 49$$

Scenario B: in **presenza** di ridondanza

Operazione 2



- Tavola degli accessi
 - Per l'entità Ristorante si effettua un accesso in lettura per leggere le stelline

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Ristorante	Entità	1	L

- Questa istruzione viene eseguita ~ 5 volte al giorno

$$TOT\ ACCESSI = 5 * n^{\circ} accesso\ in\ lettura = 5 * 1 = 5$$

Inoltre, serve calcolare il costo dal punto di vista della memoria occupata.

- Senza ridondanza: 0 byte
- Con ridondanza: ipotizziamo che l'attributo "Stelline" occupi 2 byte. Essendoci 100 ristoranti, lo spazio di memoria occupato sarà di 200 byte (0,2 Kbyte).

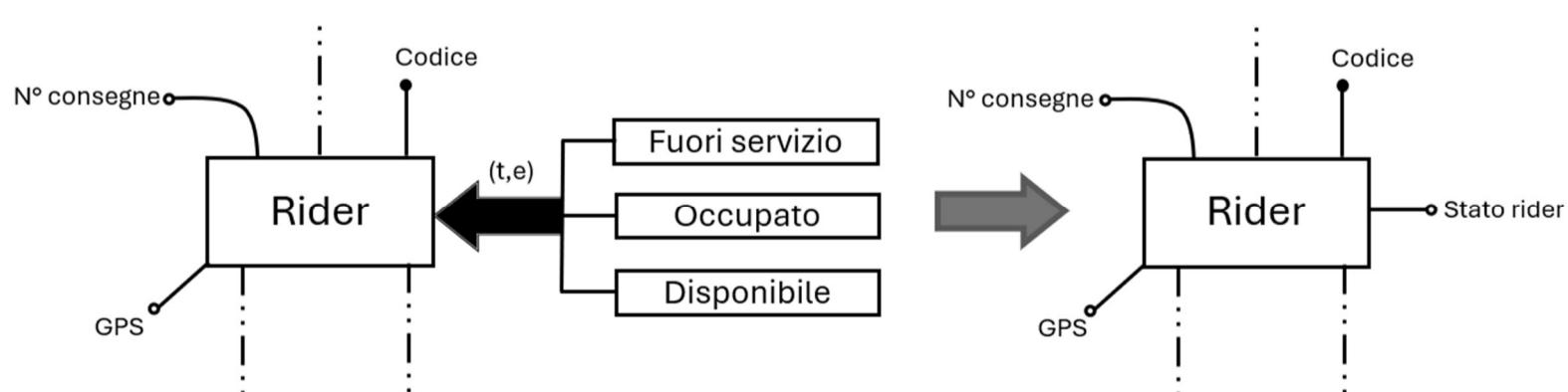
In assenza di ridondanza si effettueranno 1548 (28+1520) accessi mentre in presenza di ridondanze si effettueranno 54 (49+5) accessi. Grazie a questo calcolo si può constatare quindi che in presenza di ridondanza si effettuano meno accessi rispetto al caso in cui la ridondanza viene rimossa. Inoltre, la memoria occupata è così poca che può essere trascurata.

2.3.2. Eliminazione delle generalizzazioni

Le generalizzazioni presenti nello schema E-R sono:

- La generalizzazione su Rider riguardante lo stato di servizio.
Esso può essere di tre tipologie:
 - Fuori servizio
 - Occupato
 - Disponibile

Siccome le tre entità figlie sono prive di attributi si è utilizzata la prima strategia di eliminazione delle generalizzazioni. Essa prevede l'accorpamento delle entità figlie nelle entità genitore.



Viene aggiunto un nuovo attributo (Stato rider) che determina lo stato del rider (Fuori servizio, Occupato, Disponibile).

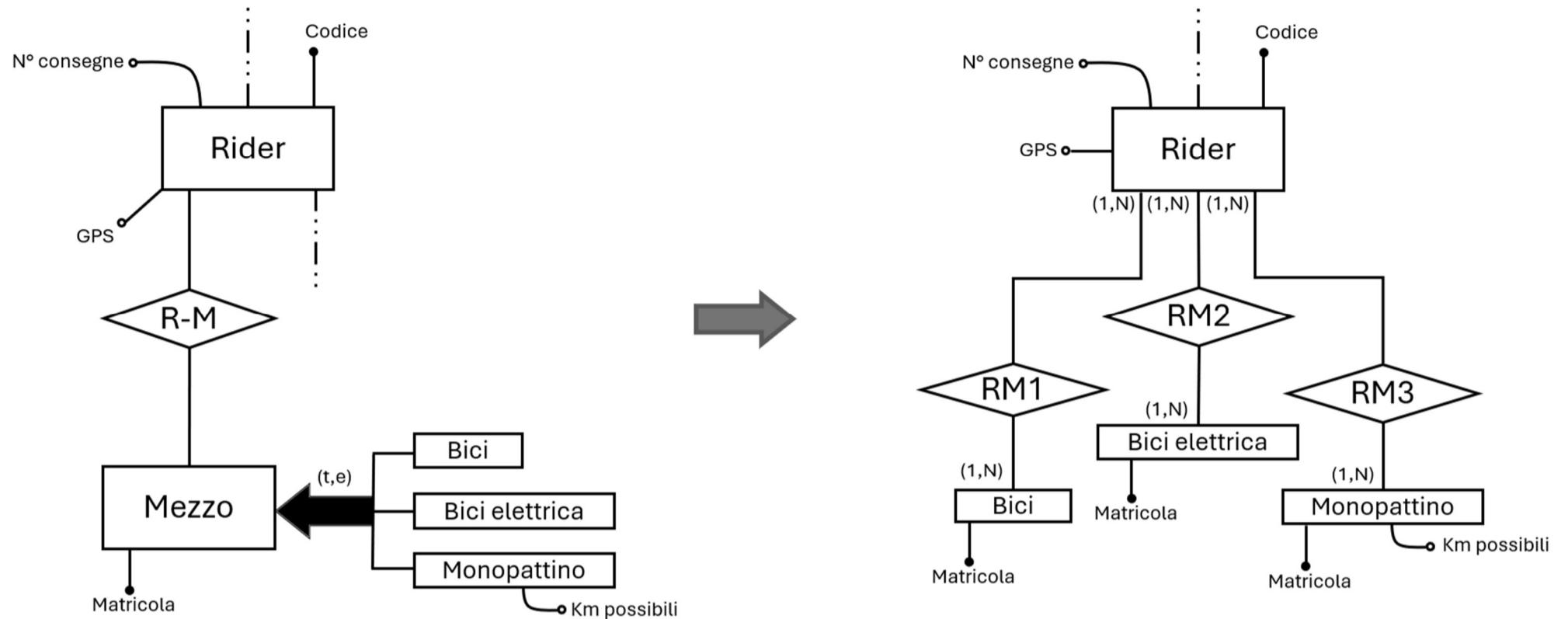
- La generalizzazione su Rider riguardante la tipologia di mezzo.

Esso può essere di tre tipologie:

- Bici
- Bici elettrica
- Monopattino

Siccome l'entità Mezzo ha un attributo identificatore (Matricola), si è utilizzata la seconda strategia di eliminazione delle generalizzazioni.

Essa prevede l'accorpamento dell'entità genitore nelle entità figlie.



Viene eliminata l'entità Mezzo e le tre entità figlie vengono legate all'entità rider tramite delle associazioni (RM1, RM2, RM3) e l'identificatore di Mezzo: Matricola, viene implementato alle tre ex entità figlie.

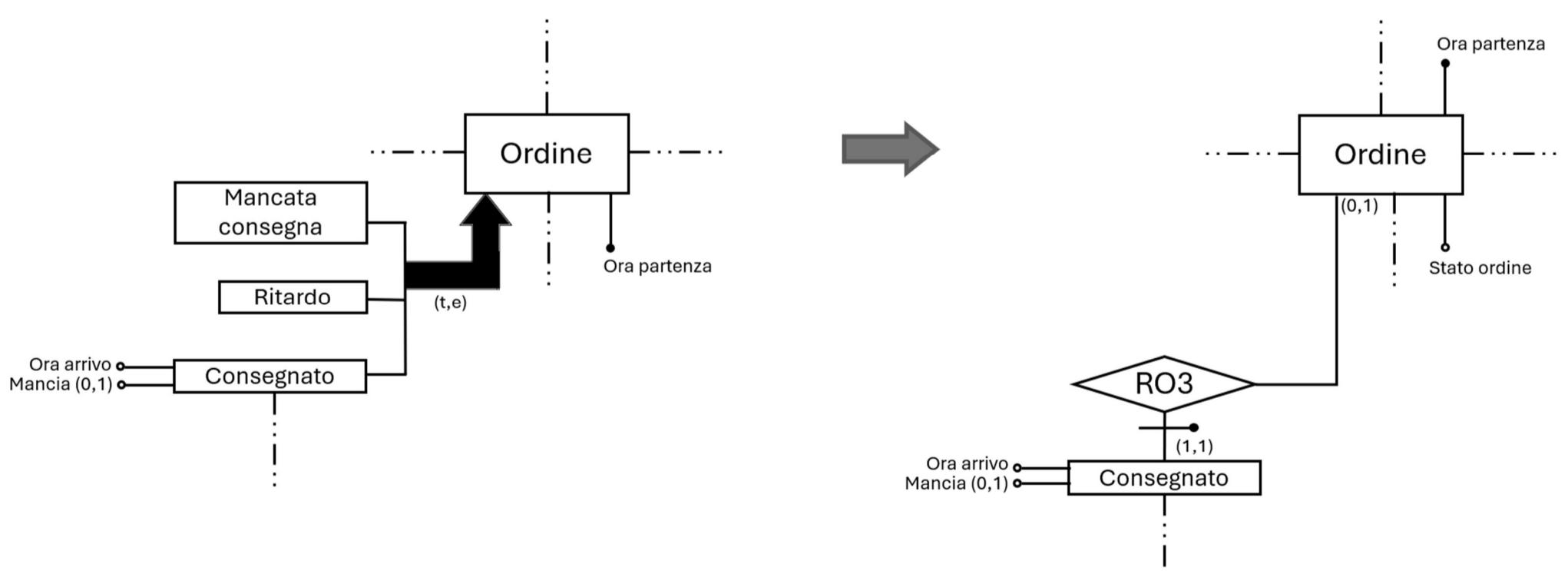
- La generalizzazione su Ordine riguardante lo stato di esso.

Esso può essere di tre tipologie:

- Mancata consegna
- Ritardo
- Consegnato

Si è utilizzata una strategia mista che sfrutta la prima e la terza strategia di eliminazione delle generalizzazioni.

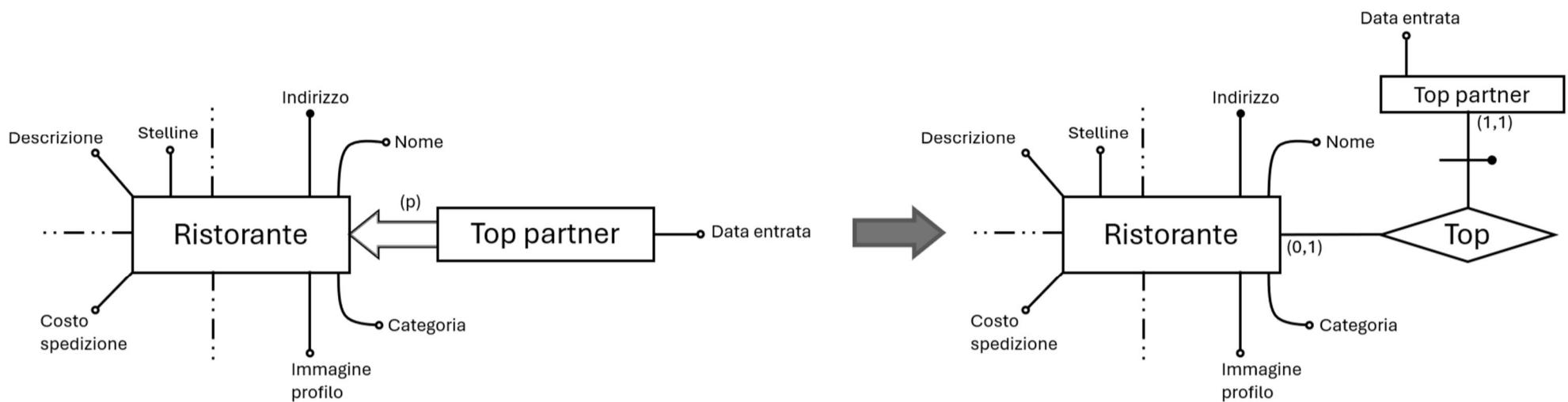
La prima prevede l'accorpamento delle entità figlie nell'entità genitore mentre la terza prevede la sostituzione della generalizzazione con delle relazioni che mantengono sia l'entità genitore sia le entità figlie.



Viene inserito un nuovo attributo ("Stato ordine") per le entità figlie prive di attributi "Mancata consegna" e "Ritardo" mentre

per l'entità figlia Consegnato viene trasformata in un'entità a parte e collegata ad ordine tramite l'associazione RO3. Consegnato manterrà i suoi attributi "Ora arrivo" e "mancia (0,1)" e l'interazione con l'associazione "Recensione consegna".

- La generalizzazione su "Ristorante" riguardante il badge di "Top partner".
Si è utilizzata la terza strategia di eliminazione delle generalizzazioni.
Essa prevede la sostituzione della generalizzazione con delle relazioni che mantengono sia l'entità genitore sia le entità figlie.

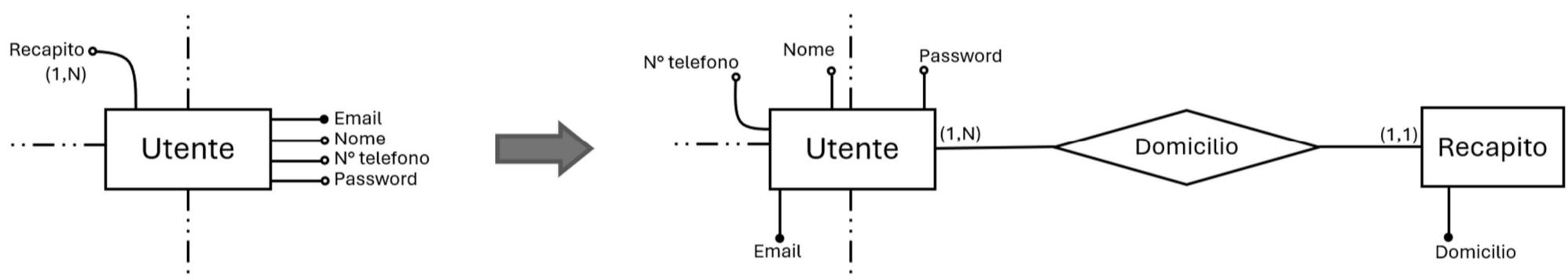


Viene aggiunta una nuova associazione (Top) che lega l'entità "Top partner".

"Top partner" manterrà il suo attributo Data entrata.

2.3.3. Eventuale partizionamento/accorpamento di entità e associazioni

L'entità "Utente", avendo un attributo multi-valore (recapito (1,N)), deve essere partizionata in quanto questo attributo non può essere tradotto in maniera naturale.



Si è introdotta una nuova associazione "Domicilio" che lega "Utente" a "Recapito": l'entità in cui viene "conservata" la via di destinazione dell'ordine.

2.3.4. Eventuale scelta degli identificatori principali

Le entità con degli identificatori esterni individuati sono:

- Borsellino
- Chat
- Piatto

In tutte e tre le entità non sono stati individuati degli identificatori semplici e definibili come tali; quindi, si è optato per dei nuovi attributi definiti ad hoc chiamati:

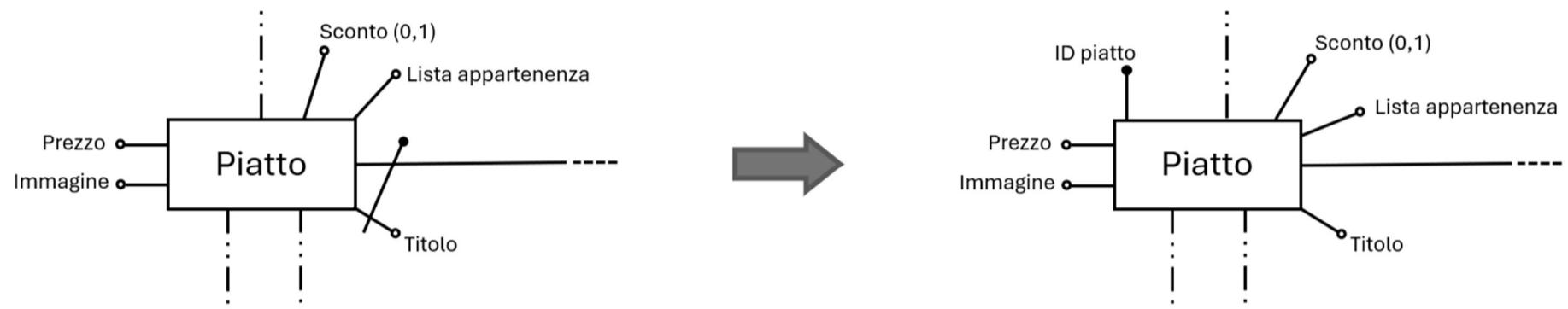
- Borsellino → ID borsellino



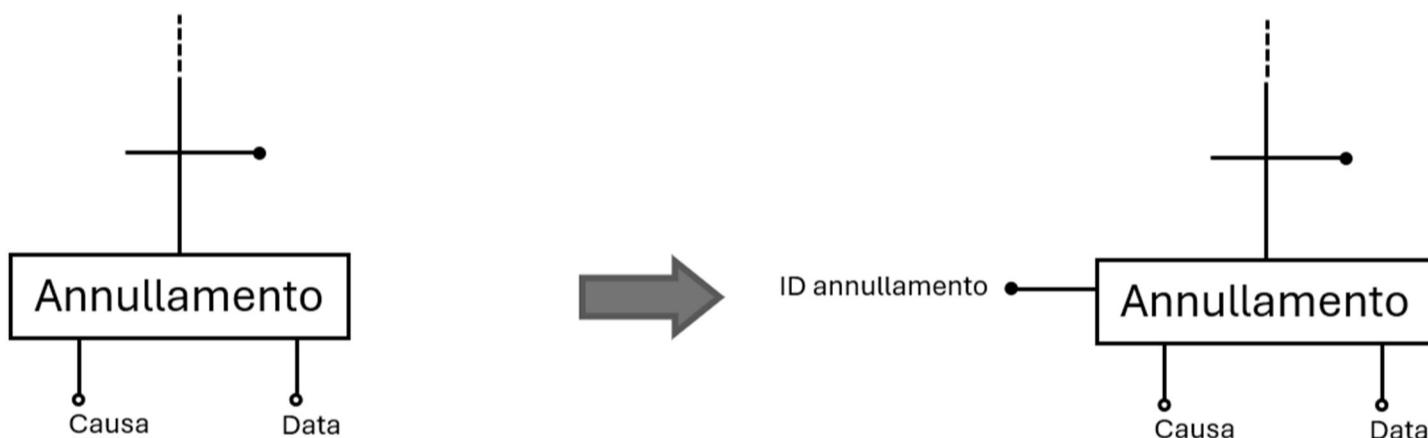
- Chat → ID chat



- Piatto → ID piatto

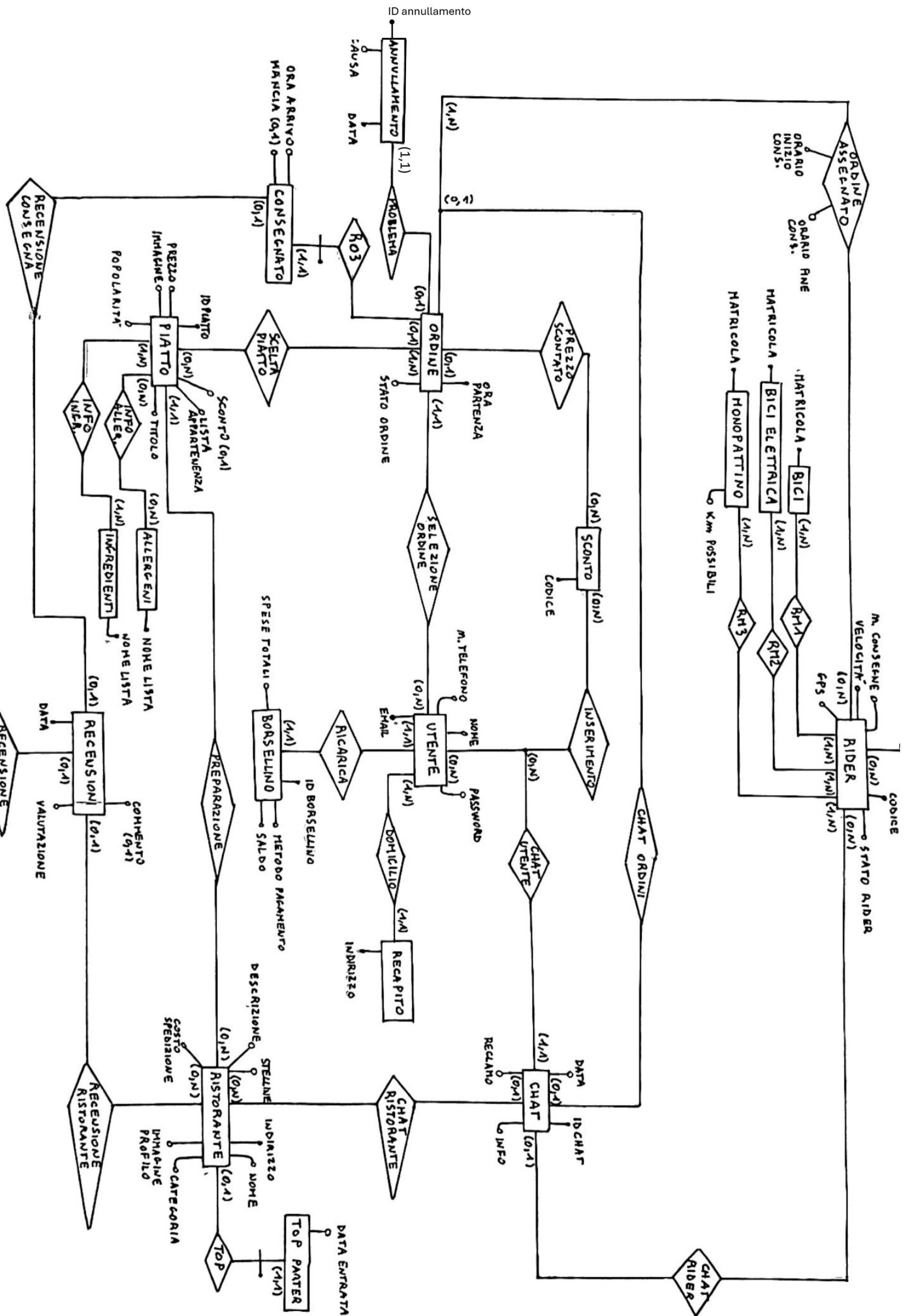


- Annullo → ID annullo



2.4. Schema E-R ristrutturato e business rules

Lo schema ristrutturato viene mostrato nella pagina successiva a causa della grandezza.



(4) GUARDARE SOPRA PER LA CONTINUAZIONE (1)

Business rules

- Dizionario dei dati sull'entità:

ENTITA'	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Utente	Individuo che usa l'app Cibora per ordinare il cibo	E-mail, Nome, n° telefono, Password	E-mail
Borsellino	Borsellino ricaricabile internamente dall'utente contenente un saldo utilizzabile per effettuare pagamenti	Saldo, Metodo pagamento, Spese totali, ID borsellino	ID borsellino
Ordine	Ordine di piatti che si vuole effettuare	Ora partenza, Stato ordine	Ora partenza
Annullamento	in caso il piatto consegnato abbia delle problematiche si può annullare la consegna specificandone la causa.	Causa, Data, ID annullamento	ID annullamento
Piatto	Piatto proposto dal ristorante e scelto dall'utente	Sconto (0,1), Immagine, Lista appartenenza, Titolo, Prezzo, Popolarità, ID piatto	ID piatto
Allergeni	Lista contenente eventuali allergeni presenti nel piatto	Nome lista	Nome lista
Ingredienti	Lista contenente gli ingredienti presenti nel piatto	Nome lista	Nome lista
Ristorante	Luogo dove vengono cucinati i piatti	Descrizione, Costo spedizione, Indirizzo, Nome, Categoria, Immagine profilo, Stelline	Indirizzo
Chat	Chat in cui utenti, ristoranti e riders possono comunicare dove l'utente, per gli ordini passati può effettuare un reclamo	Data, Reclamo, Info, ID chat	ID chat
Rider	Persona che tramite un mezzo consegna il piatto comprato dall'utente	Codice, n° consegne, GPS, Velocità, Stato rider	Codice
Sconto	Rappresenta lo sconto che un utente può utilizzare ed è accumulato in base agli ordini effettuati	Codice	Codice
Recensioni	Sezione in cui recensire la consegna, il rider o il ristorante tramite una valutazione da 1 a 5 e un commento tutti e due facoltativi.	Valutazione, Commento(0,1), Data	Data
Top partner	In base ad una buona condotta, ad un buon servizio e ad altri fattori i ristoranti possono ottenere il badge top partner per aumentare lo status quo del ristorante nell'app	Data entrata	Ristorante
Recapito	Il luogo in cui viene inviato il rider con l'ordine	Indirizzo	Indirizzo
Bici	Mezzo a pedali a due ruote utilizzato dai rider per muoversi	Matricola	Matricola
Bici elettrica	Mezzo a pedalata assistita a due ruote utilizzato dai rider per muoversi	Matricola	Matricola
Monopattino	Mezzo elettrico a due ruote utilizzato dai rider per muoversi	Matricola, Km possibili	Matricola
Consegnato	È lo stato finale, quello ottimale, in cui il piatto arriva all'utente	Ora arrivo, Mancia (0,1)	Ordine

- Dizionario dei dati sulle associazioni:

ASSOCIAZIONI	DESCRIZIONE	COMPONENTI	ATTRIBUTI
Selezione ordine	Permette di effettuare un ordine	Utente(0,N) Ordine (1,1)	
Prezzo scontato	Permette <u>applicare</u> lo sconto sull'ordine	Ordine(0,1) Sconto(0,N)	
Inserimento	Permette di <u>inserire</u> uno sconto sull'ordine	Utente(0,N) Sconto(0,N)	
Ricarica	Permette di ricaricare il saldo del borsellino	Utente(1,1) Borsellino(1,1)	
Scelta piatto	Permette di selezionare il piatto che si vuole ordinare	Ordine(1,N) Piatto(0,N)	
Preparazione	Permette di inviare l'ordine al ristorante in modo da poterlo preparare	Piatto(1,1) Ristorante(0,N)	
Info allergeni	Permette di poter leggere la lista degli allergeni	Piatto(0,N) Allergeni(0,N)	
Info ingredienti	Permette di poter leggere la lista degli ingredienti	Piatto(1,N) Ingredienti(1,N)	
Chat utente	Permette all'utente di utilizzare la chat	Utente(0,N) Chat(1,1)	
Chat ristorante	Permette al ristorante di utilizzare la chat	Ristorante(0,N) Chat(0,1)	
Chat rider	Permette al rider di utilizzare la chat	Rider(0,N) Chat(0,1)	
Chat ordini	Permette di scrivere un reclamo	Ordine(0,1) Chat(0,1)	
Ordine assegnato	Permette di assegnare l'ordine al rider	Ordine(1,N) Rider(0,N)	Orario inizio consegna, Orario fine consegna
Recensione ristorante	Permette di recensire il ristorante	Recensioni(0,1) Ristorante(0,N)	
Recensione rider	Permette di recensire il rider	Recensioni(0,1) Rider(0,N)	
Recensione consegna	Permette di recensire la consegna	Recensioni(0,1) Consegnato (0,1)	
Problema	Permette in caso di un problema dopo la consegna di annullare l'ordine	Ordine(0,1), Annullamento (0,1)	
Top	Acconsente al ristorante di poter diventare top partner	Ristorante(0,1), Top partner(1,1)	
RM1	Permette al rider di poter utilizzare la bici	Rider(1,N), Bici(1,N)	
RM2	Permette al rider di poter utilizzare la bici elettrica	Rider(1,N), Bici elettrica(1,N)	
RM3	Permette al rider di poter utilizzare il monopattino	Rider(1,N), Monopattino(1,N)	

- Vincoli di integrità:

REGOLE DI VINCOLO
(1) Lo sconto, per essere utilizzato, non deve essere usato più di una volta dallo stesso utente
(2) Il reclamo, se lo si vuole fare, deve essere effettuato solo su un ordine passato
(3) Un ristorante per essere top partner deve soddisfare diversi requisiti (vedi da riga 16 a riga 19)
(4) Un ordine Se la “posizione corrente del rider → ristorante → utente” è superiore ai 10 km deve essere selezionati solo i rider dotati di bici elettrica

(5) Un utente, se vuole, deve fare reclamo solo su ordini passati
(6) Solo l'ordine consegnato deve andare nell'entità Consegnato, gli altri rimangono nell'entità Rider

- Derivazioni:

REGOLE DI DERIVAZIONE
Il prezzo totale pagato dall'utente si ottiene... $\text{costo totale} = \text{prezzo} + \text{costo spedizione} - \text{sconto} - \text{saldo} + \text{mancia}$
Il numero di stelline si ottiene... $\text{stelline} = \frac{1}{n} \sum_{i=1}^n x_i$

2.5. Schema relazionale

Si vanno ad analizzare le varie interazioni tra entità e associazioni in modo da poter produrre delle relazioni adatte ad essere poi utilizzate per produrre il database effettivo. (N.B.: ENTITÀ – ASSOCIAZIONI).

UTENTE-RICARICA-BORSELLINO (uno a uno)

In questa associazione, essendo uno a uno, la relazione scompare e l'entità Utente acquisisce come attributo Borsellino che referenzierà l'entità BORSELLINO.

UTENTE(Email, Telefono, Nome, Password, Borsellino)
BORSELLINO(ID-borsellino, MetodoPagamento, Saldo, SpeseTotali)

- UTENTE(Borsellino) **referenzia** BORSELLINO(ID-borsellino)

UTENTE-DOMICILIO-RECAPITO (uno a molti)

In questa associazione, essendo uno a molti, l'entità RECAPITO (che ha cardinalità 1, 1) assume i valori della relazione e **referenzia** l'entità a cui è connessa ovvero: UTENTE.

RECAPITO(Indirizzo, Utente)

- RECAPITO(Utente) **referenzia** UTENTE(Email)

UTENTE-INSERIMENTO-SCONTO (molti a molti)

In questa associazione, essendo molti a molti, INSERIMENTO avrà tra gli attributi i nomi delle entità ad essa collegate e tali attributi referenzieranno gli identificatori delle entità da cui sono presi.

INSERIMENTO(Utente, Sconto)
SCONTO(Codice)

- INSERIMENTO(Utente) **referenzia** UTENTE(Email)
- INSERIMENTO(Sconto) **referenzia** SCONTO(Codice)

UTENTE-SELEZIONA ORDINE-ORDINE (uno a molti)

Essendo una 1 a molti, l'entità Ordine (che ha cardinalità 1, 1) assume i valori della relazione e **referenzia** l'altra entità, in questo caso Utente.

ORDINE(OraPartenza, StatoOrdine, Utente)

- ORDINE(Utente) **referenzia** UTENTE(Email)

SCONTO-PREZZO SCONTATO-ORDINE (uno a molti)

In questa associazione, essendo uno a molti, l'entità ORDINE (che ha cardinalità 0, 1) assume i valori della relazione e **referenzia** l'altra entità, in questo caso SCONTO.

Avendo una cardinalità (0,1), l'attributo Sconto sarà opzionale quindi verrà inserito come un attributo opzionale (attributo*).

ORDINE(OraPartenza, StatoOrdine, Utente, Sconto*)

- ORDINE(Sconto*) **referenzia SCONTODE(Codice)**

ORDINE-PROBLEMA-ANNULLAMENTO (uno a uno)

In questa associazione, essendo uno a uno, la relazione viene “eliminata” e l'entità ANNULLAMENTO acquisisce come attributi Ordine che referenzierà l'entità ORDINE.

ANNULLAMENTO(ID-annullamento, Data, Causa, Ordine)

- Annullamento(Ordine) **referenzia ORDINE(Ora-partenza)**

ORDINE-RO3-CONSEGNATO (uno a uno)

Essendoci la presenza di un identificatore esterno, semplicemente l'entità CONSEGNATO acquisisce come attributi Ordine che non solo referenzia l'entità ORDINE, ma diventa anche identificatore.

CONSEGNATO(Ordine, OraArrivo, Mancia)

- CONSEGNATO(Ordine) **referenzia ORDINE(Ora-partenza)**

ORDINE-SCELTA PIATTO-PIATTO (molti a molti)

In questa associazione, essendo molti a molti, la relazione SCELTA-PIATTO avrà tra gli attributi i nomi delle entità ad essa collegate che referenzieranno gli identificatori delle rispettive entità.

PIATTO(ID-piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, ...)

SCELTA-PIATTO(Ordine, Piatto)

- SCELTA-PIATTO(Ordine) **referenzia ORDINE(Ora-partenza)**
- SCELTA-PIATTO(Piatto) **referenzia PIATTO(ID-piatto)**

PIATTO-INFO ALLERGENI-ALLERGENI (molti a molti)

In questa associazione, essendo molti a molti, la relazione INFO-ALLERGENI avrà tra gli attributi i nomi delle entità ad essa collegate che referenzieranno gli identificatori delle rispettive entità.

ALLERGENI(NomeLista)

INFO-ALLERGENI(Piatto, Allergeni)

- INFO-ALLERGENI(Allergeni) **referenzia ALLERGENI(NomeLista)**

PIATTO-INFO INGREDIENTI-INGREDIENTI (molti a molti)

In questa associazione, essendo molti a molti, la relazione INFO-INGREDIENTI avrà tra gli attributi i nomi delle entità ad essa collegate che referenzieranno gli identificatori delle rispettive entità.

INGREDIENTI(NomeLista)

INFO-INGREDIENTI(Piatto, Ingredienti)

- INFO-INGREDIENTI(Piatto) **referenzia PIATTO(ID-piatto)**
- INFO-INGREDIENTI(Ingredienti) **referenzia INGREDIENTI(NomeLista)**

CONSEGNATO-RECENSIONE CONSEGNA-RECENSIONI (uno a uno)

In questa associazione, essendo uno a uno, le relazioni rimangono invariate; la relazione RECENSIONE-CONSEGNA ha come attributi entrambe le entità (CONSEGNATO e RECENSIONI) e le referenzia. In questo caso si è deciso di utilizzare Consegnato come identificatore mentre per quanto riguarda l'altro attributo, Recensioni, vi sarà imposto il vincolo Unique.

RECENSIONI(Data, Commento, Valutazione, Rider*, Ristorante*)

RECENSIONE-CONSEGNA(Consegnato, Recensioni)

- RECENSIONE-CONSEGNA(Consegnato) **referenzia CONSEGNATO(Ordine)**
- RECENSIONE-CONSEGNA(Recensioni) **referenzia RECENSIONI(Data)**
- Vincolo Unique su RECENSIONE-CONSEGNA(Recensioni)

RECENSIONI-RECENSIONE RIDER-RIDER (uno a molti)

In questa associazione, essendo uno a molti, l'entità RECENSIONI (con cardinalità 0, 1) referenzia la seconda entità coinvolta: Rider. Avendo cardinalità (0,1), l'attributo Rider sarà opzionale quindi verrà visualizzato come Rider*.

RIDER(Codice, NumConsegne, Velocità, GPS, Stato)

- RECENSIONI(Rider*) referenzia RIDER(Codice)

RECENSIONI-RECENSIONE RISTORANTE-RISTORANTE (uno a molti)

In questa associazione, essendo uno a molti, l'entità RECENSIONI (con cardinalità 0, 1) assumerà i valori della relazione in questione e referenzierà la seconda entità coinvolta: Ristorante.

Avendo cardinalità (0,1), l'attributo Ristorante sarà opzionale quindi verrà visualizzato come Ristorante*.

RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)

- RECENSIONI(Ristorante*) referenzia RISTORANTE(Indirizzo)

RISTORANTE-TOP-TOP PARTNER (uno a uno)

Essendoci la presenza di un identificatore esterno, semplicemente l'entità TOPPARTNER avrà tra gli attributi Ristorante che referenzia l'entità Ristorante e diventa identificatore.

TOPPARTNER(Ristorante, DataEntrata)

- TOPPARTNER(Ristorante) referenzia RISTORANTE(Indirizzo)

RISTORANTE-PREPARAZIONE-PIATTO (uno a molti)

In questa associazione, essendo uno a molti, l'entità PIATTO (con cardinalità 1, 1) assumerà i valori della relazione in questione e referenzierà la seconda entità coinvolta: RISTORANTE.

In PIATTO viene inserito Ristorante

PIATTO(ID-piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)

- PIATTO(Ristorante) referenzia RISTORANTE(Indirizzo)

CHAT-CHAT UTENTE-UTENTE (uno a molti)

In questa associazione, essendo uno a molti, l'entità CHAT (con cardinalità 1, 1) assumerà i valori della relazione in questione e referenzierà la seconda entità coinvolta: UTENTE.

CHAT(ID-chat, Data, Reclamo, Info, Utente, ..., ...)

- CHAT(Utente) referenzia UTENTE>Email)

CHAT-CHAT RISTORANTE-RISTORANTE (uno a molti)

In questa associazione, essendo uno a molti, l'entità CHAT (con cardinalità 0, 1) assumerà i valori della relazione in questione e referenzierà la seconda entità coinvolta: Ristorante.

Avendo cardinalità (0,1), l'attributo Ristorante sarà opzionale quindi verrà visualizzato come Ristorante*.

In CHAT viene aggiunto l'attributo Ristorante*

CHAT(ID-chat, Data, Reclamo, Info, Utente, Ristorante*, ...)

- CHAT(Ristorante) referenzia RISTORANTE(Indirizzo)

CHAT-CHAT RIDER-RIDER (uno a molti)

In questa associazione, essendo uno a molti, l'entità CHAT (con cardinalità 0, 1) assumerà i valori della relazione in questione e referenzierà la seconda entità coinvolta: RIDER.

Avendo cardinalità (0,1), l'attributo Ristorante sarà opzionale quindi verrà visualizzato come Ristorante*.

In CHAT viene aggiornato l'attributo Rider in Rider*

CHAT(ID-chat, Data, Reclamo, Info, Utente, Ristorante*, Rider*)

- CHAT(Rider*) referenzia RIDER(Codice)

CHAT-CHAT ORDINI-ORDINE (uno a uno)

In questa associazione, essendo uno a uno, le relazioni rimangono invariate; la relazione CHATORDINI ha come attributi entrambe le entità (CHAT e ORDINE) e le referenzia.

In questo caso si è deciso di utilizzare Ordine come identificatore mentre per quanto riguarda l'altro attributo, Chat, vi sarà imposto il vincolo Unique.

CHATORDINI(Ordine, Chat)

- CHATORDINI(Ordine) **referenzia** ORDINE(OraPartenza)
- CHATORDINI(Chat) **referenzia** CHAT(ID-chat)
- Vincolo Unique Su CHATORDINI(Chat)

RIDER-ORDINE ASSEGNATO-ORDINE (molti a molti)

In questa associazione, essendo molti a molti, la relazione ORDINEASSEGNATO ha come attributi i nomi delle entità ad essa collegate che poi sono referenziati sugli identificatori delle entità coinvolte.

ORDINEASSEGNATO(Ordine, Rider, OrarioInizioConsegna, OrarioConsegna)

- ORDINEASSEGNATO(Ordine) **referenzia** ORDINE(OraPartenza)
- ORDINEASSEGNATO(Rider) **referenzia** RIDER(Codice)

RIDER-RM1-BICI (molti a molti)

In questa associazione, essendo molti a molti, la relazione RM1 ha come attributi i nomi delle entità ad essa collegate che poi sono referenziati sugli identificatori delle entità coinvolte.

BICI(Matricola)**RM1(Bici, Rider)**

- RM1(Bici) **referenzia** BICI(Matricola)
- RM1(Rider) **referenzia** RIDER(Codice)

RIDER-RM2-BICI ELETTRICA (molti a molti)

In questa associazione, essendo molti a molti, la relazione RM2 ha come attributi i nomi delle entità ad essa collegate che poi sono referenziati sugli identificatori delle entità coinvolte.

BICIELETTRICA(Matricola)**RM2(Bici-elettrica, Rider)**

- RM2(BiciElettrica) **referenzia** BICIELETTRICA (Matricola)
- RM2(Rider) **referenzia** RIDER(Codice)

RIDER-RM3-MONOPATTINO (molti a molti)

In questa associazione, essendo molti a molti, la relazione RM3 ha come attributi i nomi delle entità ad essa collegate che poi sono referenziati sugli identificatori delle entità coinvolte.

MONOPATTINO(Matricola, KmPossibili)**RM3(Monopattino, Rider)**

- RM3(Monopattino) **referenzia** MONOPATTINO(Matricola)
- RM3(Rider) **referenzia** RIDER(Codice)

3. Implementazione

3.1. DDL di creazione del database

Per poter costruire il database serve implementare le relazioni scritte, sotto forma di schemi, nel punto 2.5. .

La struttura di una tabella in SQL è la seguente:

```
CREATE TABLE NomeTabella (
    NomeAttributo1 Dominio1 [ValoreDefault1] [Vincoli1],
    NomeAttributo2 Dominio2 [ValoreDefault2] [Vincoli2],
    ...,
    NomeAttributoN DominioN [ValoreDefaultN] [VincoliN],
);
```

Creazione della tabella BORSELLINO:

```
CREATE TABLE BORSELLINO(
    ID_borsellino varchar(100) PRIMARY KEY,
    MetodoPagamento varchar(100), --(contanti,carta,satisfpay)
    Saldo decimal(10,2),
    SpeseTotali decimal(10,2)
);
```

Creazione della tabella UTENTE:

```
CREATE TABLE UTENTE(
    Email varchar(100) PRIMARY KEY,
    Telefono varchar(13),
    Nome varchar(100) NOT NULL,
    Password varchar(100) NOT NULL,
    Borsellino varchar(100),
        FOREIGN KEY (Borsellino) REFERENCES BORSELLINO(ID_borsellino)
        ON UPDATE CASCADE
        ON DELETE SET NULL
);
```

Creazione della tabella RECAPITO:

```
CREATE TABLE RECAPITO(
    Indirizzo varchar(100) PRIMARY KEY,
    Utente varchar(100) NOT NULL,
        FOREIGN KEY (Utente) REFERENCES UTENTE>Email)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

Creazione della tabella SCONTI:

```
CREATE TABLE SCONTI(
    Codice varchar(100) PRIMARY KEY
);
```

Creazione della tabella INSERIMENTO:

```
CREATE TABLE INSERIMENTO(
    Utente varchar(100) NOT NULL,
    Sconto varchar(100) NOT NULL,
        FOREIGN KEY (Utente) REFERENCES UTENTE>Email)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Sconto) REFERENCES SCONT0(Codice)
        ON UPDATE NO ACTION
        ON DELETE SET NULL,
    PRIMARY KEY (Utente,Sconto)
);
```

Creazione della tabella ORDINE:

```
CREATE TABLE ORDINE(
    OraPartenza timestamp PRIMARY KEY,
    StatoOrdine varchar(100) NOT NULL,
    Utente varchar(100) NOT NULL,
    Sconto varchar(100),
        FOREIGN KEY (Utente) REFERENCES UTENTE>Email)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Sconto) REFERENCES SCONT0(Codice)
        ON UPDATE CASCADE
        ON DELETE SET NULL
);
```

Creazione della tabella ANNULLAMENTO:

```
CREATE TABLE ANNULLAMENTO(
    ID_annullamento varchar(100) PRIMARY KEY,
    Data timestamp NOT NULL,
    Causa varchar(100),
    Ordine timestamp NOT NULL,
        FOREIGN KEY (Ordine) REFERENCES ORDINE(OraPartenza)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

Creazione della tabella CONSEGNATO:

```
CREATE TABLE CONSEGNATO(
    Ordine timestamp PRIMARY KEY,
    OraArrivo time NOT NULL,
    Mancia decimal(5,2),
        FOREIGN KEY (Ordine) REFERENCES ORDINE(OraPartenza)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

Creazione della tabella PIATTO:

```
CREATE TABLE PIATTO (
    ID_piatto varchar(100) PRIMARY KEY,
    Prezzo decimal(10,2) NOT NULL,
    Immagine varchar(255) NOT NULL,
    Popolarità varchar(100),
    Sconto decimal(5,2),
    ListaAppartenenza varchar(255),
    Titolo varchar(100)
);
```

Creazione della tabella SCELTAPIATTO:

```
CREATE TABLE SCELTAPIATTO (
    Ordine timestamp NOT null,
    Piatto varchar(100) NOT NULL,
    FOREIGN KEY (Ordine) REFERENCES ORDINE(OraPartenza)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Piatto) REFERENCES PIATTO(ID_piatto)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Ordine, Piatto)
);
```

Creazione della tabella ALLERGENI:

```
CREATE TABLE ALLERGENI (
    NomeLista varchar(100) PRIMARY KEY
);
```

Creazione della tabella INFOALLERGENI:

```
CREATE TABLE INFOALLERGENI (
    Piatto varchar(100) NOT NULL,
    Allergeni varchar(100) NOT NULL,
    FOREIGN KEY (Piatto) REFERENCES PIATTO(ID_piatto)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Allergeni) REFERENCES ALLERGENI(NomeLista)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Piatto, Allergeni)
);
```

Creazione della tabella INGREDIENTI:

```
CREATE TABLE INGREDIENTI (
    NomeLista varchar(100) PRIMARY KEY
);
```

Creazione della tabella INFOINGREDIENTI:

```
CREATE TABLE INFOINGREDIENTI (
    Piatto varchar(100) NOT NULL,
    Ingredienti varchar(100) NOT NULL,
    FOREIGN KEY (Piatto) REFERENCES PIATTO(ID_piatto)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Ingredienti) REFERENCES INGREDIENTI(NomeLista)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Piatto, Ingredienti)

);
```

Creazione della tabella RECENSIONI:

```
CREATE TABLE RECENSIONI (
    Data date PRIMARY KEY,
    Commento varchar(255),
    Valutazione int NOT NULL
);
```

Creazione della tabella RECENSIONICONSEGNA:

```
CREATE TABLE RECENSIONICONSEGNA (
    Consegnato timestamp(2) PRIMARY KEY,
    Recensioni date,
    FOREIGN KEY (Consegnato) REFERENCES CONSEGNATO(Ordine)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Recensioni) REFERENCES RECENSIONI(Data)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    UNIQUE (Recensioni)
);
```

Creazione della tabella RIDER:

```
CREATE TABLE RIDER (
    Codice varchar(100) PRIMARY KEY,
    NumConsegne int NOT NULL,
    Velocità float NOT NULL,
    GPS varchar(100),
    Stato varchar(100)
);
```

Inserimento dell'attributo Rider in RECENSIONI:

```
/*RECENSIONI-RECENSIONE RIDER-RIDER (uno a molti)
RIDER(Codice, Num-consegne, Velocità, GPS, Stato)
• RECENSIONI(Rider*) referenzia RIDER(Codice)*/
ALTER TABLE RECENSIONI
    ADD COLUMN Rider varchar(100),
    ADD CONSTRAINT fk_rider_codice FOREIGN KEY (Rider) REFERENCES RIDER(Codice)
        ON UPDATE CASCADE
        ON DELETE SET NULL;
```

Creazione della tabella RISTORANTE:

```
CREATE TABLE RISTORANTE (
    Indirizzo varchar(255) PRIMARY KEY,
    Nome varchar(100) NOT NULL,
    Categoria varchar(100) NOT NULL,
    ImmagineProfilo varchar(255),
    CostoSpedizione decimal(3,2) NOT NULL,
    Descrizione varchar(255) NOT NULL,
    Stelline varchar(6) NOT NULL
);
```

Inserimento dell'attributo Ristorante in PIATTO:

```
/*RISTORANTE-PREPARAZIONE-PIATTO (uno a molti)
In PIATTO viene inserito Ristorante
PIATTO(Ristorante) referenzia RISTORANTE(Indirizzo)*/
ALTER TABLE PIATTO
    ADD COLUMN Ristorante varchar(100),
    ADD CONSTRAINT fk_ristorante_indirizzo
        FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(Indirizzo)
            ON UPDATE CASCADE
            ON DELETE SET NULL;
```

Inserimento dell'attributo Ristorante in RECENSIONI:

```
/*RECENSIONI-RECENSIONE RISTORANTE-RISTORANTE (uno a molti)
RISTORANTE(Indirizzo, Nome, Categoria, Immagine-profilo,
    Costo-spedizione, Descrizione, Stelline)
RECENSIONI(Ristorante*) referenzia RISTORANTE(Indirizzo)*/
ALTER TABLE RECENSIONI
    ADD COLUMN Ristorante varchar(100),
    ADD CONSTRAINT fk_recensioni_ristorante
        FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(Indirizzo)
            ON UPDATE CASCADE
            ON DELETE CASCADE;
```

Creazione della tabella TOPPARTNER:

```
CREATE TABLE TOPPARTNER (
    Ristorante varchar(100) PRIMARY KEY,
    DataEntrata date NOT NULL,
    FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(Indirizzo)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

Creazione della tabella CHAT:

```
CREATE TABLE CHAT (
    ID_chat varchar(100) PRIMARY KEY,
    Data date NOT NULL,
    Reclamo varchar(255),
    Info varchar(255),
    Utente varchar(100) NOT NULL,
    FOREIGN KEY (Utente) REFERENCES UTENTE(Email)
);
```

Inserimento dell'attributo Ristorante in CHAT:

```
/*In CHAT viene inserito l'attributo Ristorante
CHAT(ID-chat, Data, Reclamo, Info, Utente, Ristorante*, ...)
    CHAT(Ristorante) referenzia RISTORANTE(Indirizzo) */
ALTER TABLE CHAT
    ADD COLUMN Ristorante varchar(100),
    ADD CONSTRAINT fk_ristorante_indirizzo
        FOREIGN KEY (Ristorante) REFERENCES RISTORANTE(Indirizzo)
            ON UPDATE CASCADE
            ON DELETE SET NULL;
```

Inserimento dell'attributo Rider in CHAT:

```
/*In CHAT viene inserito l'attributo Rider
CHAT(ID-chat, Data, Reclamo, Info, Utente, Ristorante*, Rider*)
    CHAT(Rider*) referenzia RIDER(Codice)*/
ALTER TABLE CHAT
    ADD COLUMN Rider varchar(100),
    ADD CONSTRAINT fk_rider_codice FOREIGN KEY (Rider) REFERENCES RIDER(Codice)
        ON UPDATE CASCADE
        ON DELETE SET NULL;
```

Creazione della tabella CHATORDINI:

```
CREATE TABLE CHATORDINI (
    Ordine timestamp,
    Chat varchar(100),
    FOREIGN KEY (Ordine) REFERENCES ORDINE(OraPartenza)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Chat) REFERENCES CHAT(ID_chat)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Ordine, Chat),
    UNIQUE (Chat)
);
```

Creazione della tabella ORDINEASSEGNATO:

```
CREATE TABLE ORDINEASSEGNATO (
    Ordine timestamp,
    Rider varchar(100),
    OraInizioConsegna timestamp NOT NULL,
    OraFineConsegna timestamp,
    FOREIGN KEY (Ordine) REFERENCES ORDINE(OraPartenza)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Rider) REFERENCES RIDER(Codice)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Ordine, Rider)
);
```

Creazione della tabella BICI:

```
CREATE TABLE BICI (
    Matricola varchar(100) PRIMARY KEY
);
```

Creazione della tabella RM1:

```
CREATE TABLE RM1 (
    Bici varchar(100) NOT NULL,
    Rider varchar(100) NOT NULL,
    FOREIGN KEY (Bici) REFERENCES BICI(Matricola)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Rider) REFERENCES RIDER(Codice)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Bici, Rider)
);
```

Creazione della tabella BICIELETTRICA:

```
CREATE TABLE BICIELETTRICA (
    Matricola varchar(100) PRIMARY KEY
);
```

Creazione della tabella RM2:

```
CREATE TABLE RM2 (
    BiciElettrica varchar(100),
    Rider varchar(100),
    FOREIGN KEY (BiciElettrica) REFERENCES BICIELETTRICA(Matricola)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Rider) REFERENCES RIDER(Codice)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (BiciElettrica, Rider)
);
```

Creazione della tabella MONOPATTINO:

```
CREATE TABLE MONOPATTINO (
    Matricola varchar(100) PRIMARY KEY,
    KmPossibili decimal(10, 4) NOT NULL
);
```

Creazione della tabella RM3:

```
CREATE TABLE RM3 (
    Monopattino varchar(100),
    Rider varchar(100),
    FOREIGN KEY (Monopattino) REFERENCES MONOPATTINO(Matricola)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Rider) REFERENCES RIDER(Codice)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    PRIMARY KEY (Monopattino, Rider)
);
```

3.2. DML di popolamento di tutte le tabelle del database

Per poter inserire dei dati nelle tabelle e quindi popolare il database bisogna ovviamente rispettare i domini degli attributi e rispettare i vincoli di integrità referenziale.

La struttura per popolare una tabella in SQL è la seguente:

```
INSERT INTO NomeTabella1(NomeAttributo1, ..., NomeAttributoN) VALUES (Valore1, ..., ValoreN);
INSERT INTO NomeTabella2(NomeAttributo1, ..., NomeAttributoN) VALUES (Valore1, ..., ValoreN);
...
INSERT INTO NomeTabellaN(NomeAttributo1, ..., NomeAttributoN) VALUES (Valore1, ..., ValoreN);
```

Inserimento dei dati in BORSELLINO:

```

INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B001', 'Contanti', 34.50, 50);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B002', 'Carta', 82.30, 100);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B003', 'Contanti', 0, 15);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B004', 'Contanti', 45.5, 100);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B005', 'Satispay', 31, 24);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B006', 'Carta', 10.35, 30.44);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B007', 'Carta', 34.5, 50);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B008', 'Contanti', 10.1, 40);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B009', 'Carta', 1.10, NULL);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B010', 'Contanti', 69.90, 90.69);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B011', 'Satispay', 100.50, 140);
INSERT INTO BORSELLINO(ID_borsellino, MetodoPagamento, Saldo, SpeseTotali) VALUES ('B012', 'Satispay', 12.45, NULL);

```

Inserimento dei dati in UTENTE:

```

INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('simonepollano@gmail.com', '+393664726653', 'Simone', 'SimoneCibora', 'B001');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('kevinroberto@gmail.com', '+393313204101', 'Kevin', 'Kevin03', 'B002');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('manuelasaccone@outlook.it', '+393478659584', 'Manuela', 'eL5969%', 'B003');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('giovannipirone@libero.it', '+393804376403', 'Giovanni', 'OU:v1+4', 'B004');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('michaelgibellato@hotmail.com', '+393331146194', 'Michael', 'tb6N49~', 'B005');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('stivendededja@yahoo.it', '+393425705434', 'Stiven', 'h61C3?H', 'B006');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('beabibi04@gmail.com', '+393473336962', 'Beatrice', 'p|2P913', 'B007');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('riccardosaccone27@outlook.it', '+393342525024', 'Riccardo', 'J0t_191skls_', 'B008');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('luca.pollano65@yahoo.it', '+393316200163', 'Luca', '@=73308zr', 'B009');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('peter.parker@spiderman.com', '+393496302231', 'Peter', 'm=U~331x]', 'B010');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('giulia.cavicchia402@hotmail.com', '+393461169331', 'Giulia', 'tA5)6693', 'B011');
INSERT INTO UTENTE>Email, Telefono, Nome, Password, Borsellino
VALUES ('zuccarello.sara04@outlook.it', '+393280147464', 'Sara', 'u2,c5X4', 'B012');

```

Inserimento dei dati in RECAPITO

```

INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via piave 37', 'simonepollano@gmail.com');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via Duomo 97', 'kevinroberto@gmail.com');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Viale Ippocrate 66', 'manuelasaccone@outlook.it');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via Corio 58', 'giovannipirone@libero.it');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via Enrico Fermi 78', 'michaelgibellato@hotmail.com');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via delle Azalee 93', 'stivendededja@yahoo.it');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Piazza Pilastri 64', 'beabibi04@gmail.com');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via Vicenza 65', 'riccardosaccone27@outlook.it');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via San Pietro Ad Aram 36', 'luca.pollano65@yahoo.it');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via Vipacco 54', 'peter.parker@spiderman.com');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via torino 4/c', 'giulia.cavicchia402@hotmail.com');
INSERT INTO RECAPITO(Indirizzo, Utente) VALUES ('Via torino 6', 'zuccarello.sara04@outlook.it');

```

Inserimento dei dati in SCONTI:

```
INSERT INTO SCONTI (Codice) VALUES ('PRIMOACQUISTO');
INSERT INTO SCONTI (Codice) VALUES ('IceIce'); --gelati
INSERT INTO SCONTI (Codice) VALUES ('Hamburgerdays'); --hamburger
INSERT INTO SCONTI (Codice) VALUES ('Black Friday');
INSERT INTO SCONTI (Codice) VALUES ('p0KeY'); --pokè
INSERT INTO SCONTI (Codice) VALUES ('Supremesushi'); --sushi
INSERT INTO SCONTI (Codice) VALUES ('Italiandays'); --cibo italiano
INSERT INTO SCONTI (Codice) VALUES ('Hola Mexico'); -- cibo messicano
```

Inserimento dei dati in INSERIMENTO:

```
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('simonepollano@gmail.com', 'PRIMOACQUISTO');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('kevinroberto@gmail.com', 'PRIMOACQUISTO');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('manuelasaccone@outlook.it', 'Hamburgerdays');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('michaelgibellato@hotmail.com', 'p0KeY');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('stivendedja@yahoo.it', 'Italiandays');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('beabibi04@gmail.com', 'IceIce');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('riccardosaccone27@outlook.it', 'Italiandays');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('giulia.cavicchia402@hotmail.com', 'Black Friday');
INSERT INTO INSERIMENTO(Utente, Sconto) VALUES ('zuccarello.sara04@outlook.it', 'IceIce');
```

Inserimento dei dati in ORDINE:

```
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-05-31 23:05:00', 'Ritardo', 'simonepollano@gmail.com', 'PRIMOACQUISTO');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-09-11 15:30:50', 'Consegnato', 'kevinroberto@gmail.com', 'PRIMOACQUISTO');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-04-30 18:15:20', 'Mancata consegna', 'manuelasaccone@outlook.it', 'Hamburgerdays');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-04-15 12:00:00', 'Consegnato', 'michaelgibellato@hotmail.com', 'p0KeY');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-10-10 16:01:45', 'Consegnato', 'stivendedja@yahoo.it', 'Italiandays');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-11-16 18:50:00', 'Ritardo', 'beabibi04@gmail.com', 'IceIce');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-09-24 21:00:00', 'Consegnato', 'riccardosaccone27@outlook.it', 'Italiandays');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-08-15 13:11:12', 'Consegnato', 'giulia.cavicchia402@hotmail.com', 'Black Friday');
INSERT INTO ORDINE(OraPartenza, StatoOrdine, Utente, Sconto)
VALUES ('2024-02-19 09:10:00', 'Mancata consegna', 'zuccarello.sara04@outlook.it', 'IceIce');
```

Inserimento dei dati in ANNULLAMENTO:

```
INSERT INTO ANNULLAMENTO(ID_annullamento, Data, Causa, Ordine)
VALUES ('An01', '2024-04-30', 'Maltempo', '2024-04-30 18:15:20');
INSERT INTO ANNULLAMENTO(ID_annullamento, Data, Causa, Ordine)
VALUES ('An02', '2024-02-19', 'Incidente stradale', '2024-02-19 09:10:00');
```

Inserimento dei dati in CONSEGNATO:

```
INSERT INTO CONSEGNATO(Ordine, OraArrivo, Mancia) VALUES ('2024-09-11 15:30:50', '15:45:00', 3);
INSERT INTO CONSEGNATO(Ordine, OraArrivo, Mancia) VALUES ('2024-04-15 12:00:00', '12:15:00', NULL);
INSERT INTO CONSEGNATO(Ordine, OraArrivo, Mancia) VALUES ('2024-10-10 16:01:45', '16:10:50', 1.50);
INSERT INTO CONSEGNATO(Ordine, OraArrivo, Mancia) VALUES ('2024-09-24 21:00:00', '21:29:50', 0.50);
INSERT INTO CONSEGNATO(Ordine, OraArrivo, Mancia) VALUES ('2024-08-15 13:11:12', '13:25:30', NULL);
```

Inserimento dei dati in RISTORANTE:

```

INSERT INTO RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)
    VALUES ('Via Lagrange 17', 'La Romana', 'Gelaterie', 'Immagine profilo:...', 3.99, 'Questo ristorante è...', '5');
INSERT INTO RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)
    VALUES ('Via delle Viole 32', 'McDonalds', 'Fast food', 'Immagine profilo:...', 1.34, 'Questo ristorante è...', '4.2');
INSERT INTO RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)
    VALUES ('Via Giulio Camuzzoni 51', 'Pokè House', 'Cibo salutare', 'Immagine profilo:...', 3.00, 'Questo ristorante è...', '3.3');
INSERT INTO RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)
    VALUES ('Strada Provinciale 65 97', 'Kinshi', 'Cibo giapponese', 'Immagine profilo:...', 3.25, 'Questo ristorante è...', '2');
INSERT INTO RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)
    VALUES ('Via Nolana 47', 'Olive Garden', 'Cibo italiano', 'Immagine profilo:...', 1.50, 'Questo ristorante è...', '1');
INSERT INTO RISTORANTE(Indirizzo, Nome, Categoria, ImmagineProfilo, CostoSpedizione, Descrizione, Stelline)
    VALUES ('Via Spalato 131', 'Taco Bell', 'Cibo messicano', 'Immagine profilo:...', 0.90, 'Questo ristorante è...', '3.5');

```

Inserimento dei dati in PIATTO:

```

INSERT INTO PIATTO(ID_piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)
    VALUES ('P1111', 5.50, 'Immagine gelato al pistacchio', 'Molto popolare', NULL, 'Gelati', 'Pistacchio', 'Via Lagrange 17');
INSERT INTO PIATTO(ID_piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)
    VALUES ('P2222', 7.00, 'Immagine hamburger', 'Molto popolare', 2, 'Hamburger', 'Crispy McBacon', 'Via delle Viole 32');
INSERT INTO PIATTO(ID_piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)
    VALUES ('P3333', 13.60, 'Immagine pokè', 'Mediamente popolare', NULL, 'Pokè', 'Hawaiian Passion', 'Via Giulio Camuzzoni 51');
INSERT INTO PIATTO(ID_piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)
    VALUES ('P4444', 25, 'Immagine sushi', 'Molto popolare', 3.30, 'Sushi', 'Tris di sashimi', 'Strada Provinciale 65 97');
INSERT INTO PIATTO(ID_piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)
    VALUES ('P5555', 15.30, 'Immagine lasagna', 'Molto popolare', NULL, 'Cibo italiano', 'Lasagna al sugo', 'Via Nolana 47');
INSERT INTO PIATTO(ID_piatto, Prezzo, Immagine, Popolarità, Sconto, ListaAppartenenza, Titolo, Ristorante)
    VALUES ('P6666', 10.00, 'Immagine tacos', 'Poco popolare', NULL, 'Cibo Messicano', 'Mex tex tacos', 'Via Spalato 131');

```

Inserimento dei dati in SCELTAPIATTO:

```

INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-05-31 23:05:00', 'P1111');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-09-11 15:30:50', 'P1111');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-04-30 18:15:20', 'P3333');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-04-15 12:00:00', 'P4444');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-10-10 16:01:45', 'P2222');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-11-16 18:50:00', 'P6666');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-09-24 21:00:00', 'P3333');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-08-15 13:11:12', 'P5555');
INSERT INTO SCELTAPIATTO(Ordine, Piatto) VALUES ('2024-02-19 09:10:00', 'P6666');

```

Inserimento dei dati in ALLERGENI:

```

INSERT INTO ALLERGENI(NomeLista) VALUES ('Gelato');
INSERT INTO ALLERGENI(NomeLista) VALUES ('Hamburger');
INSERT INTO ALLERGENI(NomeLista) VALUES ('Pokè');
INSERT INTO ALLERGENI(NomeLista) VALUES ('Sushi');
INSERT INTO ALLERGENI(NomeLista) VALUES ('Lasagna');
INSERT INTO ALLERGENI(NomeLista) VALUES ('Tacos');

```

Inserimento dei dati in INFOALLERGENI:

```

INSERT INTO INFOALLERGENI(Piatto, Allergeni) VALUES ('P1111', 'Gelato');
INSERT INTO INFOALLERGENI(Piatto, Allergeni) VALUES ('P2222', 'Hamburger');
INSERT INTO INFOALLERGENI(Piatto, Allergeni) VALUES ('P3333', 'Pokè');
INSERT INTO INFOALLERGENI(Piatto, Allergeni) VALUES ('P4444', 'Sushi');
INSERT INTO INFOALLERGENI(Piatto, Allergeni) VALUES ('P5555', 'Lasagna');
INSERT INTO INFOALLERGENI(Piatto, Allergeni) VALUES ('P6666', 'Tacos');

```

Inserimento dei dati in INGREDIENTI:

```

INSERT INTO INGREDIENTI(NomeLista) VALUES ('Gelato');
INSERT INTO INGREDIENTI(NomeLista) VALUES ('Hamburger');
INSERT INTO INGREDIENTI(NomeLista) VALUES ('Pokè');
INSERT INTO INGREDIENTI(NomeLista) VALUES ('Sushi');
INSERT INTO INGREDIENTI(NomeLista) VALUES ('Lasagna');
INSERT INTO INGREDIENTI(NomeLista) VALUES ('Tacos');

```

Inserimento dei dati in INFOINGREDIENTI:

```
INSERT INTO INFOINGREDIENTI(Piatto, Ingredienti) VALUES ('P1111', 'Gelato');
INSERT INTO INFOINGREDIENTI(Piatto, Ingredienti) VALUES ('P2222', 'Hamburger');
INSERT INTO INFOINGREDIENTI(Piatto, Ingredienti) VALUES ('P3333', 'Pokè');
INSERT INTO INFOINGREDIENTI(Piatto, Ingredienti) VALUES ('P4444', 'Sushi');
INSERT INTO INFOINGREDIENTI(Piatto, Ingredienti) VALUES ('P5555', 'Lasagna');
INSERT INTO INFOINGREDIENTI(Piatto, Ingredienti) VALUES ('P6666', 'Tacos');
```

Inserimento dei dati in RIDER:

```
INSERT INTO RIDER(Codice, NumConsegne, Velocità, GPS, Stato) VALUES ('R1551', 10, 40, 'Coordinate GPS...', 'Fuori servizio');
INSERT INTO RIDER(Codice, NumConsegne, Velocità, GPS, Stato) VALUES ('R1562', 27, 25, 'Coordinate GPS...', 'Disponibile');
INSERT INTO RIDER(Codice, NumConsegne, Velocità, GPS, Stato) VALUES ('R1585', 100, 50, 'Coordinate GPS...', 'Disponibile');
INSERT INTO RIDER(Codice, NumConsegne, Velocità, GPS, Stato) VALUES ('R1533', 3, 25.5, 'Coordinate GPS...', 'Occupato');
INSERT INTO RIDER(Codice, NumConsegne, Velocità, GPS, Stato) VALUES ('R1544', 14, 33.6, 'Coordinate GPS...', 'Occupato');
INSERT INTO RIDER(Codice, NumConsegne, Velocità, GPS, Stato) VALUES ('R1600', 20, 15, 'Coordinate GPS...', 'Disponibile');
```

Inserimento dei dati in RECENSIONI:

```
INSERT INTO RECENSIONI(Data, Commento, Valutazione, Rider, Ristorante)
VALUES ('2024-09-11', 'Questo è un prodotto molto richiesto ma a mio parere per nulla buono ecc...', 2, 'R1551', 'Via Lagrange 17');
INSERT INTO RECENSIONI(Data, Commento, Valutazione, Rider, Ristorante)
VALUES ('2024-04-15', NULL, 3, 'R1562', 'Strada Provinciale 65 97');
INSERT INTO RECENSIONI(Data, Commento, Valutazione, Rider, Ristorante)
VALUES ('2024-09-24', NULL, 4, 'R1585', 'Via Giulio Camuzzoni 51');
```

Inserimento dei dati in RECENSIONICONSEGNA:

```
INSERT INTO RECENSIONICONSEGNA(Consegnato, Recensioni) VALUES ('2024-09-1 15:30:50', '2024-09-11');
INSERT INTO RECENSIONICONSEGNA(Consegnato, Recensioni) VALUES ('2024-04-15 12:00:00', '2024-04-15');
INSERT INTO RECENSIONICONSEGNA(Consegnato, Recensioni) VALUES ('2024-09-24 21:00:00', '2024-09-24');
```

Inserimento dei dati in TOPPARTNER:

```
INSERT INTO TOPPARTNER(Ristorante, DataEntrata) VALUES ('Via Lagrange 17', '2024-05-30');
INSERT INTO TOPPARTNER(Ristorante, DataEntrata) VALUES ('Via delle Viole 32', '2024-01-25');
INSERT INTO TOPPARTNER(Ristorante, DataEntrata) VALUES ('Via Nolana 47', '2024-02-17');
```

Inserimento dei dati in CHAT:

```
INSERT INTO CHAT(ID_chat, Data, Reclamo, Info, Utente, Ristorante, Rider)
VALUES ('Ch101', '2024-04-15', NULL, 'Dove sei?', 'michaelgibellato@hotmail.com', 'Via Giulio Camuzzoni 51', 'R1562');
INSERT INTO CHAT(ID_chat, Data, Reclamo, Info, Utente, Ristorante, Rider)
VALUES ('Ch022', '2024-11-16', NULL, 'Hai trovato l indirizzo?', 'beabibi04@gmail.com', 'Via Lagrange 17', 'R1585');
INSERT INTO CHAT(ID_chat, Data, Reclamo, Info, Utente, Ristorante, Rider)
VALUES ('Ch300', '2024-08-15', 'Reclamo...', NULL, 'giulia.cavicchia402@hotmail.com', 'Via delle Viole 32', 'R1600');
INSERT INTO CHAT(ID_chat, Data, Reclamo, Info, Utente, Ristorante, Rider)
VALUES ('Ch365', '2024-02-19', NULL, 'Dove sei?', 'zuccarello.sara04@outlook.it', 'Via Spalato 131', 'R1600');
```

Inserimento dei dati in CHATORDINI:

```
INSERT INTO CHATORDINI(Ordine, Chat) VALUES ('2024-08-15 13:11:12', 'Ch300');
```

Inserimento dei dati in ORDINEASSEGNATO:

```
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-05-31 23:05:00', 'R1551', '2024-05-31 23:15:00', '2024-05-31 23:45:00');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-09-11 15:30:50', 'R1585', '2024-09-11 15:30:50', '2024-09-11 15:45:00');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-04-30 18:15:20', 'R1533', '2024-04-30 18:27:00', NULL);
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-04-15 12:00:00', 'R1562', '2024-04-15 12:00:00', '2024-04-15 12:15:00');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-10-10 16:01:45', 'R1562', '2024-10-10 16:01:45', '2024-10-10 16:10:50');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-11-16 18:50:00', 'R1585', '2024-11-16 19:00:00', '2024-11-16 19:10:30');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-09-24 21:00:00', 'R1544', '2024-09-24 21:00:00', '2024-09-24 21:29:50');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-08-15 13:11:12', 'R1600', '2024-08-15 13:11:12', '2024-08-15 13:25:30');
INSERT INTO ORDINEASSEGNATO(Ordine, Rider, OraInizioConsegna, OraFineConsegna)
VALUES ('2024-02-19 09:10:00', 'R1600', '2024-02-19 09:10:00', NULL);
```

Inserimento dei dati in BICI:

```
INSERT INTO BICI(Matricola) VALUES ('Bi03011');
INSERT INTO BICI(Matricola) VALUES ('Bi04055');
```

Inserimento dei dati in RM1:

```
INSERT INTO RM1(Bici, Rider) VALUES ('Bi03011', 'R1551');
INSERT INTO RM1(Bici, Rider) VALUES ('Bi04055', 'R1562');
```

Inserimento dei dati in BICIELETTRICA:

```
INSERT INTO BICIELETTRICA(Matricola) VALUES ('BiI20801');
INSERT INTO BICIELETTRICA(Matricola) VALUES ('BiI18001');
```

Inserimento dei dati in RM2:

```
INSERT INTO RM2(BiciElettrica, Rider) VALUES ('BiI20801', 'R1585');
INSERT INTO RM2(BiciElettrica, Rider) VALUES ('BiI18001', 'R1533');
```

Inserimento dei dati in MONOPATTINO:

```
INSERT INTO MONOPATTINO(Matricola, KmPossibili) VALUES ('Mo34259', 100.50);
INSERT INTO MONOPATTINO(Matricola, KmPossibili) VALUES ('Mo52316', 30.25);
```

Inserimento dei dati in RM3:

```
INSERT INTO RM3(Monopattino, Rider) VALUES ('Mo34259', 'R1544');
INSERT INTO RM3(Monopattino, Rider) VALUES ('Mo52316', 'R1600');
```

3.3. Operazioni di cancellazione e modifica per verificare i vincoli e gli effetti causati dalle operazioni su chiavi esterne

Per controllare se i vincoli sono corretti si fanno ad effettuare dei test di UPDATE e DELETE sulle tabelle.

Con il test di UPDATE si vuole andare ad aggiornare un valore della tabella (che sia vincolato) e si verifica se viene aggiornato correttamente oppure no.

```
UPDATE NomeTabella SET NomeAttributo = 'xyz' WHERE NomeAttributo = 'abc';
```

Con il test di DELETE si vuole andare ad eliminare un valore della tabella (che sia vincolato) e si verifica se viene eliminato correttamente oppure no.

```
DELETE FROM NomeTabella WHERE NomeAttributo = 'xyz';
```

Le tabelle su cui sono stati controllati i vincoli sono le seguenti:

- UTENTE
- PIATTO
- RM1

UTENTE:

La tabella visualizzata è la seguente:

	email [PK] character varying (100)	telefono character varying (13)	nome character varying (100)	password character varying (100)	borsellino character varying (100)
1	simonepollano@gmail.com	+393664726653	Simone	SimoneCibora	B001
2	kevinroberto@gmail.com	+393313204101	Kevin	Kevin03	B002
3	manuelasaccone@outlook.it	+393478659584	Manuela	eL5969%	B003
4	giovannipirone@libero.it	+393804376403	Giovanni	0U:v1+4	B004
5	michaelgibellato@hotmail.com	+393331146194	Michael	tb6N49~	B005
6	stivendedja@yahoo.it	+393425705434	Stiven	h61C3?H	B006
7	beabibi04@gmail.com	+393473336962	Beatrice	p 2P913	B007
8	riccardosaccone27@outlook.it	+393342525024	Riccardo	J0t_191skls_	B008
9	luca.pollano65@yahoo.it	+393316200163	Luca	@=73308zr	B009
10	peter.parker@spiderman.com	+393496302231	Peter	m=U~331x]	B010
11	giulia.cavicchia402@hotmail.com	+393461169331	Giulia	tA5)6693	B011
12	zuccarello.sara04@outlook.it	+393280147464	Sara	u2,c5X4	B012

L'operazione di UPDATE viene effettuata sull'attributo ID_borsellino.

Quello che si va ad effettuare è modificare il valore di ID_borsellino della tupla con tale valore uguale a B001...

Nella creazione della tabella si è stabilito che in caso di comandi di UPDATE si aggiornino tutti quei valori inerenti allo stesso utente (CASCADE)...

```
-- Modifica della chiave primaria 'B001' in 'B110'
UPDATE BORSELLINO SET ID_borsellino = 'B110' WHERE ID_borsellino = 'B001';
```

	email [PK] character varying (100)	telefono character varying (13)	nome character varying (100)	password character varying (100)	borsellino character varying (100)
1	kevinroberto@gmail.com	+393313204101	Kevin	Kevin03	B002
2	manuelasaccone@outlook.it	+393478659584	Manuela	eL5969%	B003
3	giovannipirone@libero.it	+393804376403	Giovanni	0U:v1+4	B004
4	michaelgibellato@hotmail.com	+393331146194	Michael	tb6N49~	B005
5	stivendedja@yahoo.it	+393425705434	Stiven	h61C3?H	B006
6	beabibi04@gmail.com	+393473336962	Beatrice	p 2P913	B007
7	riccardosaccone27@outlook.it	+393342525024	Riccardo	J0t_191skls_	B008
8	luca.pollano65@yahoo.it	+393316200163	Luca	@=73308zr	B009
9	peter.parker@spiderman.com	+393496302231	Peter	m=U~331x]	B010
10	giulia.cavicchia402@hotmail.com	+393461169331	Giulia	tA5)6693	B011
11	zuccarello.sara04@outlook.it	+393280147464	Sara	u2,c5X4	B012
12	simonepollano@gmail.com	+393664726653	Simone	SimoneCibora	B110

L'operazione di DELETE viene effettuata sull'attributo ID_borsellino.

Quello che si va ad effettuare è eliminare il valore di ID_borsellino della tupla con tale valore uguale a B002; questa, una volta eliminata mantiene tutte le altre tuple "attive" mantenendo come valore NULL.

Viene inserito il valore NULL perché nella creazione della tabella si è stabilito che in caso di DELETE del valore viene inserito il valore nullo (SET NULL)...

```
-- Cancellazione della chiave primaria 'B002'
DELETE FROM BORSELLINO WHERE ID_borsellino = 'B002';
```

	email [PK] character varying (100)	telefono character varying (13)	nome character varying (100)	password character varying (100)	borsellino character varying (100)
1	manuelasaccone@outlook.it	+393478659584	Manuela	eL5969%	B003
2	giovannipirone@libero.it	+393804376403	Giovanni	0U:v1+4	B004
3	michaelgibellato@hotmail.com	+393331146194	Michael	tb6N49~	B005
4	stivededja@yahoo.it	+393425705434	Stiven	h61C3?H	B006
5	beabibi04@gmail.com	+393473336962	Beatrice	p 2P913	B007
6	riccardosaccone27@outlook.it	+393342525024	Riccardo	J0t_191skls_	B008
7	luca.pollano65@yahoo.it	+393316200163	Luca	@=73308zr	B009
8	peter.parker@spiderman.com	+393496302231	Peter	m=U~331x]	B010
9	giulia.cavicchia402@hotmail.com	+393461169331	Giulia	tA5)6693	B011
10	zuccarello.sara04@outlook.it	+393280147464	Sara	u2,c5X4	B012
11	simonepollano@gmail.com	+393664726653	Simone	SimoneCibora	B110
12	kevinroberto@gmail.com	+393313204101	Kevin	Kevin03	[null]

Nella tabella seguente (BORSELLINO) si può vedere come non esista più il borsellino dell'utente con ID_borsellino uguale a B002 ed il borsellino con ID precedente uguale a B001 abbia un ID uguale a R110.

	id_borsellino [PK] character varying (100)	metodopagamento character varying (100)	saldo numeric (10,2)	spesetotali numeric (10,2)
2	B004	Contanti	45.50	100.00
3	B005	Satispay	31.00	24.00
4	B006	Carta	10.35	30.44
5	B007	Carta	34.50	50.00
6	B008	Contanti	10.10	40.00
7	B009	Carta	1.10	[null]
8	B010	Contanti	69.90	90.69
9	B011	Satispay	100.50	140.00
10	B012	Satispay	12.45	[null]
11	B110	Contanti	34.50	50.00

PIATTO:

La tabella visualizzata è la seguente:

	id_piatto [PK] character varying (100)	prezzo numeric (10,2)	immagine character varying (255)	popolarità character varying (100)	sconto numeric (5,2)	listaappartenenza character varying (255)	titolo character varying (100)	ristorante character varying (100)
1	P1111	5.50	Immagine gelato al pistacchio	Molto popolare	[null]	Gelati	Pistacchio	Via Lagrange 17
2	P2222	7.00	Immagine hamburger	Molto popolare	2.00	Hamburger	Cripsi McBacon	Via delle Viole 32
3	P3333	13.60	Immagine pokè	Mediamente popolare	[null]	Pokè	Hawaiian Passion	Via Giulio Camuzzoni 51
4	P4444	25.00	Immagine sushi	Molto popolare	3.30	Sushi	Tris di sashimi	Strada Provinciale 65 97
5	P5555	15.30	Immagine lasagna	Molto popolare	[null]	Cibo italiano	Lasagna al sugo	Via Nolana 47
6	P6666	10.00	Immagine tacos	Poco popolare	[null]	Cibo Messicano	Mex tex tacos	Via Spalato 131

L'operazione di UPDATE viene effettuata sull'attributo Ristorante.

Quello che si va ad effettuare è modificare il valore di Ristorante della tupla con tale valore uguale a Via Lagrange 17...

```
-- Modifica dell'indirizzo 'Via Lagrange 17' in 'Via Roma 33'
UPDATE RISTORANTE SET Indirizzo = 'Via Roma 33' WHERE Indirizzo = 'Via Lagrange 17';
```

	id_piatto [PK] character varying (100)	prezzo numeric (10,2)	immagine character varying (255)	popolarità character varying (100)	sconto numeric (5,2)	listaappartenenza character varying (255)	titolo character varying (100)	ristorante character varying (100)
1	P2222	7.00	Immagine hamburger	Molto popolare	2.00	Hamburger	Crispy McBacon	Via delle Viole 32
2	P3333	13.60	Immagine pokè	Mediamente popolare	[null]	Pokè	Hawaiian Passion	Via Giulio Camuzzoni 51
3	P4444	25.00	Immagine sushi	Molto popolare	3.30	Sushi	Tris di sashimi	Strada Provinciale 65 97
4	P5555	15.30	Immagine lasagna	Molto popolare	[null]	Cibo italiano	Lasagna al sugo	Via Nolana 47
5	P6666	10.00	Immagine tacos	Poco popolare	[null]	Cibo Messicano	Mex tex tacos	Via Spalato 131
6	P1111	5.50	Immagine gelato al pistacchio	Molto popolare	[null]	Gelati	Pistacchio	Via Roma 33

L'operazione di DELETE viene effettuata sull'attributo Ristorante.

Quello che si va ad effettuare è eliminare il valore di Ristorante della tupla con tale valore uguale a Via Spalato 131; questa, una volta eliminata mantiene tutte le altre tuple "attive" mantenendo come valore NULL.

Viene inserito il valore NULL perché nella creazione della tabella si è stabilito che in caso di DELETE del valore viene inserito il valore nullo (SET NULL)...

-- Cancellazione della riga con indirizzo 'Via Spalato 131'

DELETE FROM RISTORANTE WHERE Indirizzo = 'Via Spalato 131';

	id_piatto [PK] character varying (100)	prezzo numeric (10,2)	immagine character varying (255)	popolarità character varying (100)	sconto numeric (5,2)	listaappartenenza character varying (255)	titolo character varying (100)	ristorante character varying (100)
1	P2222	7.00	Immagine hamburger	Molto popolare	2.00	Hamburger	Crispy McBacon	Via delle Viole 32
2	P3333	13.60	Immagine pokè	Mediamente popolare	[null]	Pokè	Hawaiian Passion	Via Giulio Camuzzoni 51
3	P4444	25.00	Immagine sushi	Molto popolare	3.30	Sushi	Tris di sashimi	Strada Provinciale 65 97
4	P5555	15.30	Immagine lasagna	Molto popolare	[null]	Cibo italiano	Lasagna al sugo	Via Nolana 47
5	P1111	5.50	Immagine gelato al pistacchio	Molto popolare	[null]	Gelati	Pistacchio	Via Roma 33
6	P6666	10.00	Immagine tacos	Poco popolare	[null]	Cibo Messicano	Mex tex tacos	[null]

Nella tabella seguente (CHAT) si può vedere come nonostante l'indirizzo "Via Spalato 131" è stato eliminato la chat avviata tra l'utente e il ristorante è ancora "salvata" e che il ristorante con indirizzo precedente uguale a "Via Lagrange 17" abbia un indirizzo uguale a "Via Roma 33".

	id_chat [PK] character varying (100)	data date	reclamo character varying (255)	info character varying (255)	utente character varying (100)	ristorante character varying (100)	rider character varying (100)
1	Ch300	2024-08-15	Reclamo:...	[null]	giulia.cavicchia402@hotmail.com	Via delle Viole 32	R1600
2	Ch022	2024-11-16	[null]	Hai trovato l'indirizzo?	beabibi04@gmail.com	Via Roma 33	R1585
3	Ch365	2024-02-19	[null]	Dove sei?	zuccarello.sara04@outlook.it	[null]	R1600
4	Ch101	2024-04-15	[null]	Dove sei?	michaelgibellato@hotmail.com	Via Giulio Camuzzoni 51	[null]

RM1:

La tabella visualizzata è la seguente:

	bici [PK] character varying (100)	rider [PK] character varying (100)
1	Bi03011	R1551
2	Bi04055	R1562

L'operazione di UPDATE viene effettuata sull'attributo Rider.

Quello che si va ad effettuare è modificare il valore di Rider della tupla con tale valore uguale a R1551...

Nella creazione della tabella si è stabilito che in caso di comandi di UPDATE si aggiornino tutti quei valori inerenti allo stesso utente (CASCADE)...

-- Modifica del codice 'R1551' in 'R2222'

UPDATE RIDER SET Codice = 'R2222' WHERE Codice = 'R1551';

	bici [PK] character varying (100)	rider [PK] character varying (100)
1	Bi04055	R1562
2	Bi03011	R2222

L'operazione di DELETE viene effettuata sull'attributo Rider.

Quello che si va ad effettuare è eliminare il valore di Rider della tupla con tale valore uguale a R1562.

Nella creazione della tabella si è stabilito che in caso di comandi di DELETE si elimina tutti quei valori inerenti allo stesso utente (CASCADE)...

```
-- Cancellazione della riga con codice 'R1562'
DELETE FROM RIDER WHERE Codice = 'R1562';
```

The diagram illustrates the deletion of a row from the RIDER table. It consists of two tables, one above the other, connected by a downward-pointing arrow. Both tables have columns 'bici' [PK] character varying (100) and 'rider' [PK] character varying (100). In the top table, there are two rows: row 2 with bici 'Bi03011' and rider 'R2222', and row 1 with bici 'Bi03011' and rider 'R1562'. The row with rider 'R1562' is highlighted with a dashed border. The bottom table shows the result after the deletion: only the row with bici 'Bi03011' and rider 'R2222' remains.

	bici [PK] character varying (100)	rider [PK] character varying (100)
2	Bi03011	R2222
1	Bi03011	R1562

	bici [PK] character varying (100)	rider [PK] character varying (100)
1	Bi03011	R2222

Nella tabella seguente (RIDER) si può vedere come non esista più il rider con codice R1562 ed il rider con codice precedente uguale a R1551 abbia codice uguale a R2222.

	codice [PK] character varying (100)	numconsegne integer	velocità double precision	gps character varying (100)	stato character varying (100)
1	R1585	100	50	Coordinate GPS:...	Disponibile
2	R1533	3	25.5	Coordinate GPS:...	Occupato
3	R1544	14	33.6	Coordinate GPS:...	Occupato
4	R1600	20	15	Coordinate GPS:...	Disponibile
5	R2222	10	40	Coordinate GPS:...	Fuori servizio