

Laboratorio Basi di Dati 2023/24

Progetto di piattaforma di Food Delivery Cibora

Progetto svolto da:

Nicolas Fiocco

Matricola 1052526

Giorgio Francesco Florida

Matricola 1054584

Tiziano Jhonny Floriddia

Matricola 1041726

Schema guida

1. [Progettazione concettuale](#)
 - 1.1 [Requisiti iniziali](#)
 - 1.2 [Glossario dei termini](#)
 - 1.3 [Requisiti rivisti e strutturati in gruppi di frasi omogenee](#)
 - 1.3.1 [Ristrutturazione dei requisiti](#)
 - 1.3.2 [Requisiti ristrutturati](#)
 - 1.3.3 [Requisiti strutturati in gruppi di frasi omogenee](#)
 - 1.3.4 [Specifiche sulle operazioni](#)
 - 1.4 [Schema E-R + business rules](#)
 - 1.4.1 [Schema E-R](#)
 - 1.4.2 [Dizionario dei dati \(entità\)](#)
 - 1.4.3 [Dizionario dei dati \(relazioni\)](#)
 - 1.4.4 [Vincoli d'integrità](#)
 - 1.4.5 [Derivazioni](#)
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 [Eliminazione delle generalizzazioni](#)
 - 2.3.3 [Eventuale partizionamento/accorpamento di entità e associazioni](#)
 - 2.3.4 [Eventuale scelta degli identificatori principali](#)
 - 2.4 [Schema E-R ristrutturato + business rules](#)
 - 2.4.1 [Schema E-R ristrutturato](#)
 - 2.4.2 [Dizionario dei dati \(entità\)](#)
 - 2.4.3 [Dizionario dei dati \(relazioni\)](#)
 - 2.4.4 [Vincoli d'integrità](#)
 - 2.4.5 [Derivazioni](#)
 - 2.5 [Schema relazionale + vincoli di integrità referenziale](#)

2.5.1 [Schema relazionale](#)

2.5.2 [Vincoli di integrità referenziale](#)

3. [Implementazione](#)

3.1 [DDL di creazione del database](#)

3.2 [DML di popolamento di tutte le tabelle del database](#)

3.3 [Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne](#)

3.3.1 [Operazioni di cancellazione](#)

3.3.2 [Operazioni di modifica](#)

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, email, password, numero di telefono, indirizzo di recapito. Una volta registrati, **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** può ricaricare il proprio **borsellino** in qualsiasi momento. Inoltre, gli **utenti** possono sottoscrivere la modalità **premium** che garantisce una priorità sugli **ordini**.

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 **pianze** 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**.

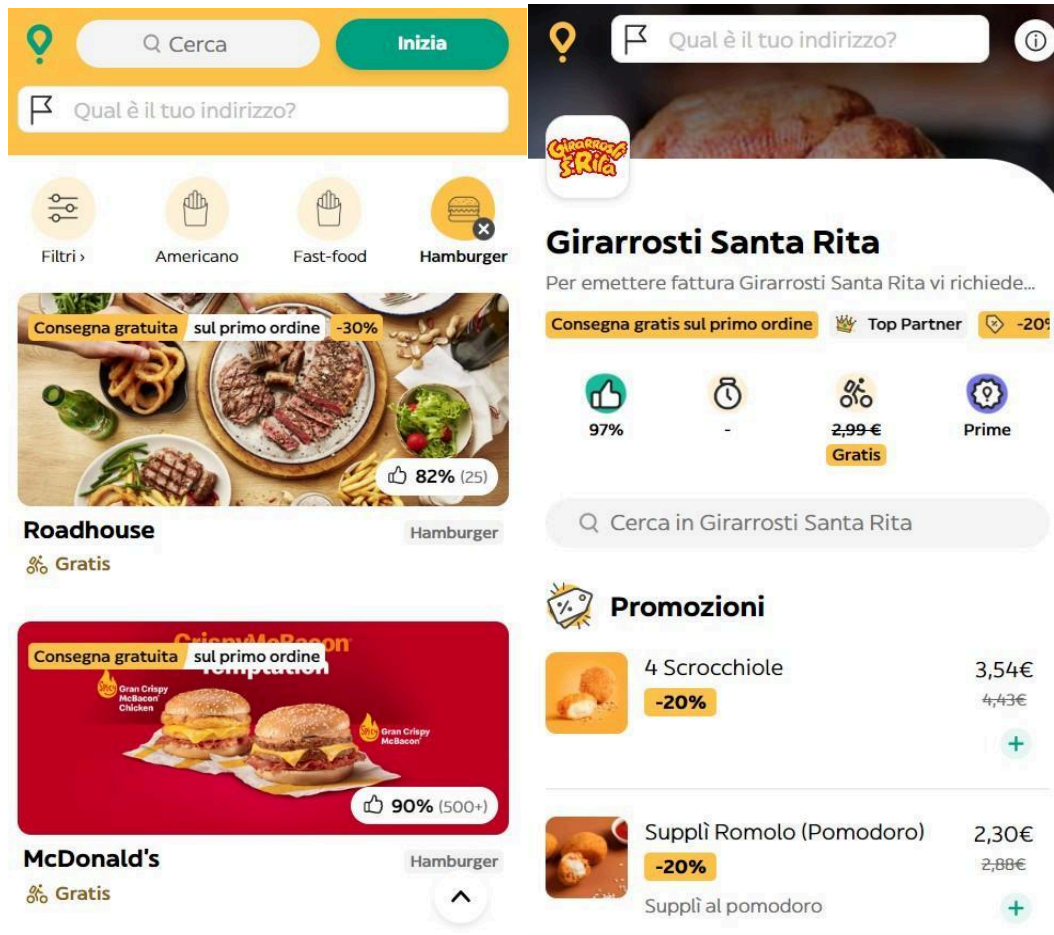


Figura 1 (a) La lista dei **ristoranti** con filtro "Hamburger". (b) I dettagli di un **ristorante**.

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

Al momento dell'**ordine**, il sistema trova il **rider** libero con la somma minima della distanza dal **ristorante** più la distanza dall'**utente**. Tuttavia, per **ordini** che prevedano un tragitto "posizione

corrente del **rider**-> **ristorante**-> **cliente**" superiore ai 10 km, solo i **rider** con bici elettrica vengono interpellati. Per monitorare le prestazioni dei **ciclofattorini**, si vuole tenere traccia del numero di **consegne** effettuate da ognuno, del momento in cui il **cibo** da **consegnare** viene affidato ad un **rider** e, per le **consegne** già completate, anche dell'ora in cui **l'ordine** è stato recapitato al **cliente**.

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

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

Una volta al mese, vengono aggiornate le seguenti classifiche:

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

1.2 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Utente	Persona registrata alla piattaforma	Cliente	Ordine, Borsellino, Chat, Ordine Cancellato, Recensione
Utente premium	<u>Tipo particolare di utente</u> che ha sottoscritto un abbonamento	Utente	Utente, Ordine, Borsellino, Chat, Ordine Cancellato, Recensione
Ristorante	Fornitore del servizio		Piatto, Chat Ristorante, Ordine Cancellato, Recensione
Top Partner	<u>Tipo particolare di ristorante</u>	Ristorante	Ristorante, Piatto, Chat Ristorante, Ordine Cancellato, Recensione
Rider	Individuo adibito alle consegne	Fattorino, ciclofattorino	Chat Rider, Consegna, Recensione
Rider-bicicletta	<u>Tipo particolare di rider</u> che consegna con la bicicletta	Fattorino, ciclofattorino	Rider, Chat Rider, Consegna, Recensione
Rider-bicicletta elettrica	<u>Tipo particolare di rider</u> che consegna con la bicicletta-elettrica	Fattorino, ciclofattorino	Rider, Chat Rider, Consegna, Recensione
Rider-monopattino	<u>Tipo particolare di rider</u> che consegna con il monopattino	Fattorino, ciclofattorino	Rider, Chat Rider, Consegna, Recensione

Rider Disponibile	<u>Tipo particolare di rider</u> che è disponibile per la consegna	Fattorino, ciclofattorino	Rider, Chat Rider, Consegna, Recensione
Rider FuoriServizio	<u>Tipo particolare di rider</u> non disponibile a nessuna consegna	Fattorino, ciclofattorino	Rider, Chat Rider, Consegna, Recensione
Rider Occupato	<u>Tipo particolare di rider</u> occupato in una consegna	Fattorino, ciclofattorino	Rider, Chat Rider, Consegna, Recensione
Recensione	Numero di stelle e commento (facoltativo)	Valutazione	Consegna Effettuata, Utente, Ristorante, Rider
Recensione Rider	<u>Tipo particolare di recensione</u> effettuata al rider del servizio	Valutazione	Recensione, Consegna Effettuata, Utente, Rider
Recensione Ristorante	<u>Tipo particolare di recensione</u> effettuata al ristorante del servizio	Valutazione	Recensione, Consegna Effettuata, Utente, Ristorante
Ordine	Insieme di piatti scelti dall'utente		Piatto, Utente
Ordine Accettato	<u>Tipo particolare di ordine</u> accettato dal ristorante		Ordine, Piatto, Utente, Codici Sconto, Consegna
Ordine Cancellato	<u>Tipo particolare di ordine</u> cancellato da utente o ristorante		Ordine, Piatto, Utente, Ristorante
Piatto	Servizio offerto	Cibo, portata, pietanze	Ristorante, Ordine

Piatto dolce	<u>Tipo particolare di piatto</u> dolce		Piatto, Ristorante, Ordine
Piatto salato	<u>Tipo particolare di piatto</u> salato		Piatto, Ristorante, Ordine
Piatto più venduto	<u>Tipo particolare di piatto</u> più venduto		Piatto, Ristorante, Ordine
Piatto in promozione	<u>Tipo particolare di piatto</u> in promozione		Piatto, Ristorante, Ordine
Chat	Spazio virtuale dove è possibile lo scambio di messaggi tra utente, rider e ristorante		Consegna in Ritardo, Consegna non Effettuata, Utente
Chat-Ristorante	<u>Tipo particolare di chat</u> con il ristorante		Chat, Consegna in Ritardo, Utente, Ristorante
Chat-Rider	<u>Tipo particolare di chat</u> con il rider		Chat, Consegna in Ritardo, Utente, Rider
Codici di sconto	Promozioni ottenibili da ordini effettuati e accettati		Borsellino, Ordine Accettato

Consegna	Ordine accettato pronto ad essere consegnato		Ordine Accettato, Rider Disponibile
Consegna effettuata	<u>Tipo particolare di consegna</u> recapitata		Consegna, Ordine Accettato, Rider Disponibile
Consegna in ritardo	<u>Tipo particolare di consegna</u> non in orario		Consegna, Ordine Accettato, Rider Disponibile
Consegna non effettuata	<u>Tipo particolare di consegna</u> non recapitata		Consegna, Ordine Accettato, Rider Disponibile
Borsellino	Portafoglio online ricaricabile che permette di effettuare i pagamenti degli ordini		Utente, Codici sconto

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

1.3.1 Ristrutturazione dei requisiti

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~~ **rider** che effettuano le consegne ~~in bicicletta~~ **con un mezzo di trasporto**.

Fasi dell'analisi dei requisiti

- Individuare sinonimi/omonimi e unificare i termini
- Rendere esplicito il riferimento tra termini
- Scegliere il corretto livello di astrazione
- Standardizzare la struttura delle frasi

~~Per beneficiare del servizio, ogni utente deve registrarsi inserendo~~ **Per l'utente registrato** **rappresentiamo** nome, email, 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 può ricaricare il proprio borsellino in qualsiasi momento. Inoltre, gli utenti ~~possono sottoscrivere~~ **che sottoscrivono** la modalità premium ~~che garantisce~~ **hanno** una priorità sugli ordini.

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~~ **un** nome, ~~una~~ **la** descrizione, ~~un~~ **l'**indirizzo, il costo della ~~spedizione~~ **consegna**, ~~un~~ **l'**immagine di profilo e ~~un numero di stelline~~ **la valutazione in base** **quinaria** aggiornato ogni lunedì sulla base ~~della percentuale di recensioni positive~~ **della media delle** **recensioni del ristorante** dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al tipo di ~~e~~ **piatti** offerto (ad esempio: fast food, vegetariano, ...).

I ristoranti che ~~dimostrano di saper garantire~~ **garantiscono** un ottimo servizio (almeno 20 ordini consegnati correttamente **dal ristorante**, una valutazione ~~clienti~~ **utenti** 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 **da parte degli utenti** del 2.5%) ~~sono considerati~~ **diventano** 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 **del Top Partner**, aiutando ad aumentare la credibilità **del ristorante** e ottenere la fiducia dei ~~clienti~~ **utenti**. Per i Top Partner si ~~vuole tenere traccia della~~ **rappresenta la** data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti una lista di piatti da ordinare. Ogni ~~portata~~ **piatto** ~~ha~~ **è** **rappresentato dal** ~~un~~ **un** titolo, ~~un~~ **l'**immagine, ~~una~~ **la** lista di ingredienti, ~~una~~ **la** 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~~ **piatti** ed effettuare l'ordine **dal ristorante**. **Gli ordini** Finché non sono affidati ad un rider per la consegna, ~~gli ordini~~ possono essere annullati sia ~~dai clienti~~ **dagli utenti**, sia ~~dai ristoratori~~ **dai ristoranti**. Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami **riguardo l'ordine** inviando un messaggio al ristorante.

Il sistema gestisce un numero arbitrario di riders dove **per** ogni rider ~~è identificato da un~~ **rappresentiamo il** 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~~ **indicano** quanti km possono effettuare prima che si scarichi la batteria **del monopattino**.

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 ~~ciclofattori~~ **rider**, si ~~vole~~ **tiene** traccia del numero di consegne effettuate ~~da ognuno~~ **dai rider**, del momento in cui il cibo da consegnare viene affidato ~~ad un~~ **al** rider e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato ~~al cliente~~ **all'utente**.

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

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

Inoltre è anche presente la possibilità **per l'utente** 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~~ **Piatti** più popolari
- Ristoranti con più recensioni positive
- ~~Clienti~~ **Utenti** che hanno speso di più

1.3.2 Requisiti ristrutturati

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 con un mezzo di trasporto.

Per l'utente registrato rappresentiamo nome, email, password, numero di telefono, indirizzo di recapito. 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 può ricaricare il proprio borsellino in qualsiasi momento. Inoltre, gli utenti che sottoscrivono la modalità premium hanno una priorità sugli ordini.

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 dal nome, la descrizione, l'indirizzo, il costo della consegna, l'immagine di profilo e la valutazione in base quinquaria aggiornata ogni lunedì sulla base della media delle recensioni del ristorante. Ogni ristorante appartiene a una o più categorie in base al tipo di piatti offerto (ad esempio: fast food, vegetariano, ...).

I ristoranti che garantiscono un ottimo servizio (almeno 20 ordini consegnati correttamente dal ristorante, una valutazione utenti 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 da parte degli utenti del 2.5%) diventano Top Partner. I Top Partner compaiono in sezioni dedicate all'interno dell'app mobile Cibora e ricevono uno speciale badge che attesta il servizio eccellente del Top Partner, aiutando ad aumentare la credibilità del ristorante e ottenere la fiducia degli utenti. Per i Top Partner si rappresenta la data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti una lista di piatti da ordinare. Ogni piatto è rappresentato dal titolo, l'immagine, la lista di ingredienti, la 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 piatti ed effettuare l'ordine dal ristorante. Gli ordini finché non sono affidati ad un rider per la consegna possono essere annullati sia dagli utenti, sia dai ristoranti. Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami riguardo l'ordine inviando un messaggio al ristorante.

Il sistema gestisce un numero arbitrario di riders dove per ogni rider rappresentiamo il codice, lo stato (occupato/disponibile/fuori servizio), la 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 indicano quanti km possono effettuare prima che si scarichi la batteria del monopattino.

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 rider, si tiene traccia del numero di consegne effettuate dai rider, del momento in cui il cibo da consegnare viene affidato al rider e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato all'utente.

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

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

Inoltre è anche presente la possibilità per l'utente 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
- Piatti più popolari
- Ristoranti con più recensioni positive
- Utenti che hanno speso di più

1.3.3 Requisiti strutturati in gruppi di frasi omogenee

Frasi di carattere generale	<ul style="list-style-type: none">● Si deve progettare la base di dati per Cibora, 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 con un mezzo di trasporto.
Frasi relative agli utenti	<ul style="list-style-type: none">● Per l'utente registrato rappresentiamo<ul style="list-style-type: none">○ nome○ email○ password○ numero di telefono○ indirizzo di recapito● 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 può ricaricare il proprio borsellino in qualsiasi momento. Inoltre, gli utenti che sottoscrivono la modalità premium hanno una priorità sugli ordini.● Ogni utente può selezionare una lista di piatti ed effettuare l'ordine dal ristorante.● L'utente può collezionare codici di sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.

Frase relative ai rider	<ul style="list-style-type: none"> ● Il sistema gestisce un numero arbitrario di riders dove per ogni rider rappresentiamo: <ul style="list-style-type: none"> ○ il codice ○ lo stato (occupato/disponibile/fuori servizio) ○ la 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 indicano quanti km possono effettuare prima che si scarichi la batteria del monopattino. ● Al rider con la somma minima della distanza dal cliente e dal ristorante, viene assegnato l'ordine. Se questa distanza è maggiore di 10 km, vengono interpellati solo i rider che hanno la bici elettrica. ● Per monitorare le prestazioni dei rider, si tiene traccia del numero di consegne effettuate dai rider, del momento in cui il cibo da consegnare viene affidato al rider e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato all'utente.
Frase relative ai piatti	<ul style="list-style-type: none"> ● Ogni piatto è rappresentato da: <ul style="list-style-type: none"> ○ titolo ○ l'immagine ○ la lista di ingredienti ○ la 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.).

Frase relative ai ristoranti	<ul style="list-style-type: none"> ● Ogni ristorante è rappresentato dal: <ul style="list-style-type: none"> ○ nome ○ la descrizione ○ l'indirizzo ○ il costo della consegna ○ l'immagine di profilo ○ la valutazione in base quinquaria aggiornata ogni lunedì sulla base della media delle recensioni del ristorante ● Ogni ristorante appartiene a una o più categorie in base al tipo di piatti offerto (ad esempio: fast food, vegetariano, ...). ● I ristoranti propongono agli utenti una lista di piatti da ordinare.
Frase relative alle valutazioni	<ul style="list-style-type: none"> ● Una volta che l'ordine viene consegnato, l'utente ha la possibilità di recensire il ristorante e il rider. Questa valutazione ha una scala da 1 a 5 stelle, il commento testuale è facoltativo.
Frase relative a tipi specifici di ristoranti	<ul style="list-style-type: none"> ● I ristoranti che garantiscono un ottimo servizio (almeno 20 ordini consegnati correttamente dal ristorante, una valutazione utenti 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 da parte degli utenti del 2.5%) diventano Top Partner. I Top Partner compaiono in sezioni dedicate all'interno dell'app mobile Cibora e ricevono uno speciale badge che attesta il servizio eccellente del Top Partner, aiutando ad aumentare la credibilità del ristorante e ottenere la fiducia degli utenti. Per i Top Partner si rappresenta la data in cui sono entrati a far parte della categoria.

Frase relative agli ordini	<ul style="list-style-type: none"> • Gli ordini finché non sono affidati ad un rider per la consegna possono essere annullati sia dagli utenti, sia dai ristoranti. Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami riguardo l'ordine inviando un messaggio al ristorante.
Frase relative al borsellino	<ul style="list-style-type: none"> • Il borsellino ha un saldo che viene aggiornato ad ogni ordinazione e l'utente può ricaricare il proprio borsellino in qualsiasi momento.
Frase relative alle operazioni	<ul style="list-style-type: none"> • La base di dati deve supportare le seguenti operazioni: <ul style="list-style-type: none"> ○ Riders più veloci nel consegnare gli ordini, aggiornato ogni mese ○ Piatti più popolari, aggiornato ogni mese ○ Ristoranti con più recensioni positive, aggiornato ogni mese ○ Utenti che hanno speso di più, aggiornato ogni mese ○ Controllare le condizioni per la qualifica di Top Partner, aggiornato ogni lunedì ○ Aggiornamento valutazione media delle stelle dei ristoranti, aggiornato ogni lunedì ○ Controllare la condizione di Utente premium e aggiornare lo status degli utenti, aggiornato ad un mese dalla data di pagamento

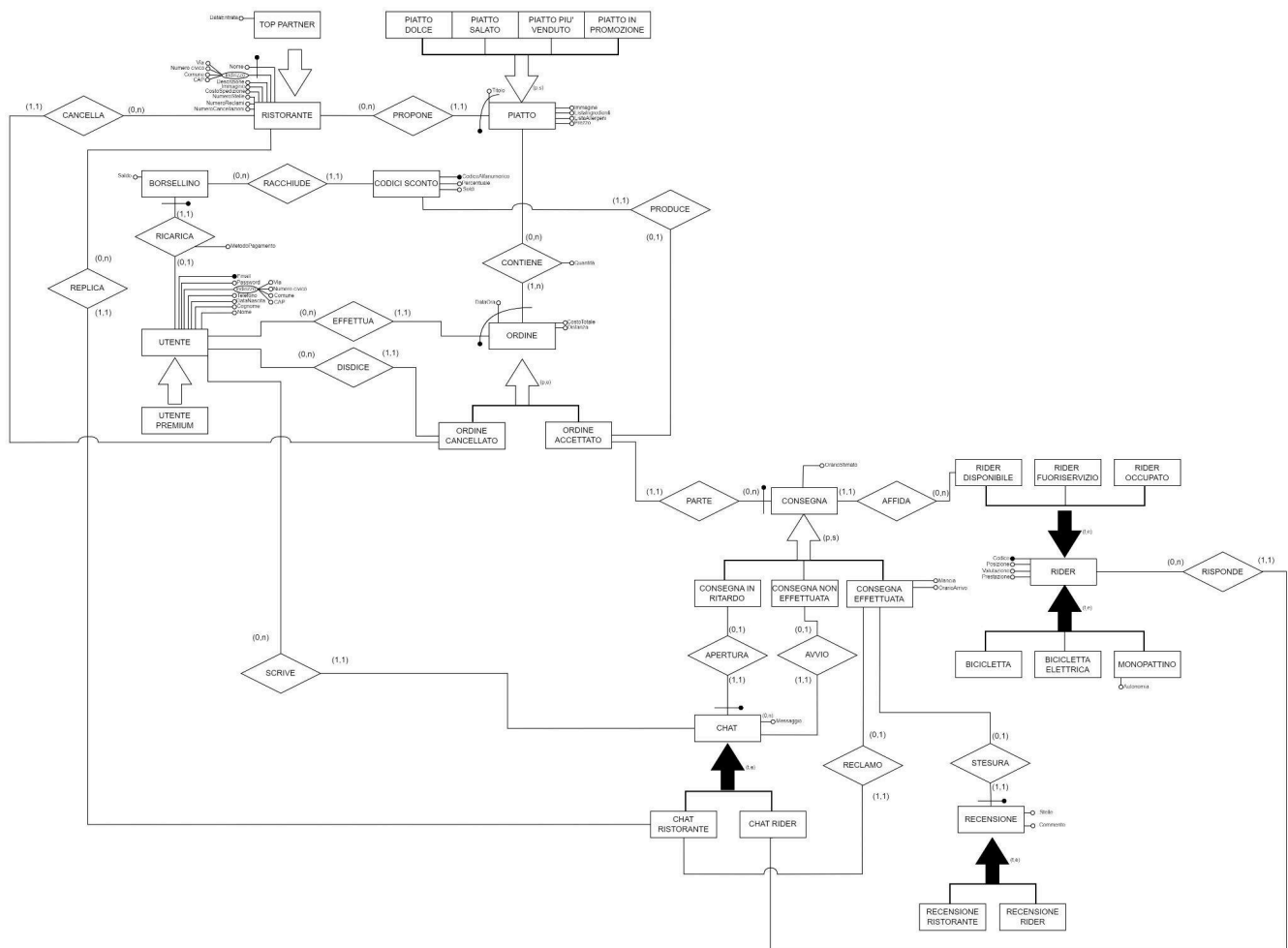
1.3.4 Specifiche sulle operazioni

Operazione	Descrizione	Frequenza
Operazione 1	Effettuare un calcolo per aggiornare la classifica dei rider più veloci	1 volta / mese
Operazione 2	Effettuare una classifica dei cibi più popolari	1 volta / mese
Operazione 3	Effettuare una classifica dei ristoranti con più recensioni positive	1 volta / mese
Operazione 4	Calcolare la classifica dei clienti che hanno speso di più	1 volta / mese
Operazione 5	Controllare le condizioni per la qualifica di Top Partner e	1 volta / giorno

	aggiornare lo status dei partner.	
Operazione 6	Aggiornare la valutazione media dei ristoranti	1 volta / giorno
Operazione 7	Controllare la condizione di Utente premium e aggiornare lo status degli utenti	1 volta / giorno

1.4 Schema E-R + business rules

1.4.1 Schema E-R



1.4.2 Dizionario dei dati (entità)

Entità	Descrizione	Attributi	Identificatore
Utente	Utente del servizio	Email, Password, Nome,Cognome, DataNascita, Telefono, Indirizzo	Email
Utente premium	<u>Entità figlia di utente</u> Utente che ha pagato per l'abbonamento premium		Email
Ristorante	Ristorante del servizio	Nome, Descrizione, Indirizzo, CostoSpedizione, Immagine, NumeroStelle	Nome, Indirizzo
Top Partner	<u>Entità figlia di ristorante</u> Ristorante con un badge identificativo	DataEntrata	Nome, Indirizzo
Rider	Rider del servizio	Codice, Posizione, Valutazione, Prestazione	Codice
Rider Occupato	<u>Entità figlia di rider</u> Rider già impegnato in un'altra consegna		
Rider Fuori Servizio	<u>Entità figlia di rider</u> Rider che ha finito il turno di lavoro		
Rider Disponibile	<u>Entità figlia di rider</u> Rider pronto alla consegna		
Bicicletta	<u>Entità figlia di rider</u> Rider con bicicletta		
Monopattino	<u>Entità figlia di rider</u> Rider con monopattino	Autonomia	
Bicicletta Elettrica	<u>Entità figlia di rider</u> Rider con bicicletta elettrica		

Ordine	Insieme dei piatti scelti dall'utente	DataOra, CostoTotale, Distanza	Utente, DataOra, Piatto
Ordine Accettato	<u>Entità figlia di ordine</u> Ordine che non è stato rifiutato e che viene preparato per la consegna		
Ordine Cancellato	<u>Entità figlia di ordine</u> Ordine cancellato dall'utente o dal ristorante		
Consegna	Ordine che viene affidato al rider	OrarioStimato	Ordine Accettato
Consegna in ritardo	<u>Entità figlia di consegna</u> Consegna in ritardo		
Consegna effettuata	<u>Entità figlia di consegna</u> Consegna effettuata	Mancia, OrarioArrivo	
Consegna non effettuata	<u>Entità figlia di consegna</u> Consegna non effettuata		
Piatto	Piatto proposto dal ristorante	Titolo, Immagine, ListaIngredienti(0,n), ListaAllergeni(0,n), Prezzo	Titolo, Ristorante
Dolce	<u>Entità figlia di piatto</u> Piatto dolce		
Salato	<u>Entità figlia di piatto</u> Piatto salato		
Più venduto	<u>Entità figlia di piatto</u> Piatto più venduto		

Promozione	<u>Entità figlia di piatto</u> Piatto in promozione		
Borsellino	Salvadanaio digitale dell'utente	Saldo	Utente
Recensione	Recensione dell'utente	Stelle, Commento	Consegna effettuata
Recensione Rider	<u>Entità figlia di recensione</u> Recensione assegnata a rider		
Recensione Ristorante	<u>Entità figlia di recensione</u> Recensione assegnata a ristorante		
Chat	Spazio virtuale dove è possibile lo scambio di messaggi	Messaggio(0,n)	Consegna in ritardo
Chat Rider	<u>Entità figlia di chat</u> Messaggi tra utente e rider		
Chat Ristorante	<u>Entità figlia di chat</u> Messaggi tra utente e ristorante		
Codice Sconto	Codice che permette di ottenere uno sconto	CodiceAlfanumerico, Percentuale, Soldi	CodiceAlfanumerico

1.4.3 Dizionario dei dati (relazioni)

Relazioni	Descrizione	Componenti	Attributi
Disdice	Disdetta dell'ordine da parte dell'utente	Utente (0,n), Ordine Cancellato (1,1)	
Cancella	Cancellazione dell'ordine da parte del ristorante	Ristorante (0,n), Ordine Cancellato (1,1)	
Ricarica	Ricarica del borsellino	Utente (0,1), Borsellino (1,1)	Metodo di pagamento
Racchiude	Codici sconto racchiusi nel borsellino	Borsellino (0,n), Codici sconto (1,1)	
Effettua	Ordine effettuato da un utente	Utente (0,n), Ordine (1,1)	
Contiene	Piatto contenuto nell'ordine	Ordine (1,n), Piatto (0,n)	Quantità
Propone	Piatti proposti dal ristorante	Ristorante (0,n), Piatto (1,1)	
Parte	Ordine accettato che parte per la consegna	Ordine Accettato(1,1), Consegna(0,n)	
Stesura	Stesura di una recensione	Consegna effettuata(0,1), Recensione (1,1)	
Scrive	Scrittura di un messaggio	Utente (0,n), Chat (1,1)	
Replica	Replica ad un messaggio dal ristorante	Ristorante (0,n), Chat Ristorante (1,1)	
Risponde	Risposta ad un messaggio dal rider	Rider (0,n), Chat Rider (1,1)	
Apertura	Apertura della chat tramite consegna in ritardo	Consegna in ritardo (0,1), Chat (1,1)	
Produce	Codici sconto prodotti dall'ordine	Ordine Accettato (0,1), Codici sconto (1,1)	
Affida	Consegna affidata al rider	Consegna(1,1), Rider disponibile(0,n)	
Avvia	Chat avviata in seguito di un ordine non consegnato correttamente	Consegna non effettuata(0,1), Chat(1,1)	
Reclamo	Reclamo che può fare l'utente se c'è un problema con la consegna effettuata	Consegna effettuata(0,1), Chat Ristorante(1,1)	

1.4.4 Vincoli d'integrità

1. Ogni ristorante **deve** avere una valutazione quinaria aggiornata ogni lunedì sulla base della media delle proprie recensioni.
2. Il ristorante Top Partner **deve** avere consegnato correttamente almeno 20 ordini.
3. Il ristorante Top Partner **deve** avere una valutazione utenti maggiore o uguale a 4.5 stelline su cinque.
4. Il ristorante Top Partner **deve** avere una percentuale massima di ordini annullati dal ristorante dell'1.5% e una percentuale massima di ordini con reclami da parte degli utenti del 2.5%.
5. La recensione **deve** diventare visibile quando sia host che ospite hanno fatto la recensione
6. Gli ordini **devono** poter essere annullati sia dall'utente che dal ristorante fino al momento della consegna dell'ordine al rider.
7. L'utente può recensire sia il rider che il ristorante con una valutazione che **deve** essere compresa tra 1 e 5.

1.4.5 Derivazioni

1. La valutazione del ristorante **si ottiene** facendo la media tra le valutazioni ricevute dagli utenti che hanno effettuato un ordine.
2. La valutazione del rider **si ottiene** facendo la media tra le valutazioni ricevute dagli utenti che hanno ricevuto l'ordine.
3. La percentuale di ordini annullati **si ottiene** dal rapporto degli ordini annullati con il totale degli ordini effettuati.
4. La percentuale di ordini con reclami **si ottiene** dal rapporto di ordini che hanno riscontrato problemi con il numero di ordini consegnati.
5. Il numero di recensioni del ristorante **si ottiene** conteggiando le recensioni riguardanti gli ordini consegnati che fanno riferimento a quel determinato ordine.
6. Il costo totale **si ottiene** sommando i costi dei piatti ordinati con il prezzo della consegna.
7. La prestazione del rider **si ottiene** sottraendo il numero di ordini consegnati con i minuti di ritardo totali e sottraendo ulteriormente il numero di ordini non consegnati * 2, la moltiplicazione per 2 è un modo per dare un peso maggiore agli ordini non consegnati rispetto a quelli consegnati.

2. Progettazione logica

2.1 Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	10 000 000
Utente premium	E	1 000 000
Ristorante	E	200 000
Top Partner	E	20 000
Rider	E	500 000
Rider Occupato	E	150 000
Rider Fuori Servizio	E	50 000
Rider Disponibile	E	300 000
Rider Bicicletta	E	200 000
Rider Monopattino	E	100 000
Rider Bicicletta Elettrica	E	200 000
Ordine	E	250 000 000
Ordine Accettato	E	225 000 000
Ordine Cancellato	E	25 000 000
Piatto	E	10 000 000
Dolce	E	5 000 000
Salato	E	5 000 000
Più venduto	E	250 000
Promozione	E	200 000
Borsellino	E	9 500 000
Consegna	E	225 000 000
Consegna in ritardo	E	25 000 000

Consegna effettuata	E	200 000 000
Consegna non effettuata	E	10 000 000
Recensione	E	22 500 000
Recensione Rider	E	10 000 000
Recensione Ristorante	E	12 500 000
Chat	E	35 000 000
Chat Rider	E	17 500 000
Chat Ristorante	E	17 500 000
Codice Sconto	E	25 000 000
Cancella	A	8 250 000
Disdice	A	16 750 000
Ricarica	A	9 500 000
Racchiude	A	25 000 000
Effettua	A	250 000 000
Contiene	A	750 000 000
Propone	A	10 000 000
Parte	A	225 000 000
Stesura	A	22 500 000
Scrive	A	150 000 000
Replica	A	49 500 000
Risponde	A	30 000 000
Produce	A	25 000 000
Affida	A	225 000 000
Avvia	A	25 000 000
Apertura	A	10 000 000
Reclamo	A	4 000 000

Motivazioni:

- Abbiamo pensato che un valore verosimile per il numeri di utenti del servizio potesse essere 10 milioni.
- Abbiamo supposto che 1/10 degli utenti siano premium (e quindi che abbiano sottoscritto un abbonamento), quindi 1 milione.
- Abbiamo supposto che i ristoranti siano 200 mila.
- Abbiamo supposto che i ristoranti Top Partner siano 1/10 dei ristoranti totali, ovvero 20 mila.

- Abbiamo pensato che un valore verosimile per il numero di rider potesse essere 500 mila, di cui 200 mila di rider con bicicletta, 200 mila di rider con bicicletta elettrica e 100 mila di rider con monopattino, inoltre abbiamo ipotizzato che ci siano 50 mila rider fuori servizio, 150 mila occupati e 300 mila disponibili.
- Abbiamo ipotizzato che ci siano 250 milioni di ordini, di cui la maggioranza (225 milioni) ordini accettati e i restanti 25 milioni sono ordini cancellati.
- Consegna avrà lo stesso volume di ordini accettati. Di questi 225 milioni, 200 milioni sono consegne effettuate correttamente, 10 milioni sono ordini non consegnati, 25 milioni sono consegne in ritardo
- Supponendo che i ristoranti in media propongono una media di 50 piatti, l'entità piatto avrà un valore di 10 milioni, categorizzata attraverso le seguenti:
dolce (5 milioni), salato (5 milioni), più venduto (250 mila), promozione (200 mila).
- Supponiamo che non tutti gli utenti registrati (10 milioni) inseriscono un metodo di pagamento, quindi sul totale degli utenti avremmo 9 milioni e mezzo di borsellini.
- Abbiamo ipotizzato che ci siano 22 milioni di recensioni, di cui 10 milioni per la recensione sul rider e 12 milioni per la recensione sul ristorante, poiché una parte degli utenti, una volta che gli viene consegnato l'ordine, non lascia una recensione.
- Abbiamo ipotizzato che ci siano 35 milioni di chat, suddivise in chat rider (17 milioni e mezzo) e chat ristorante (17 milioni e mezzo) per restare coerenti al motivo per il quale si apre la chat, cioè, consegna in ritardo o consegna non effettuata.
- Per i codici sconto abbiamo ipotizzato che ne venga creato 1 per ogni 9 ordini, quindi il valore sarà 25 milioni .
- Cancelli avrà $\frac{1}{3}$ del volume di ordine cancellato.
- Disdice avrà $\frac{2}{3}$ del volume di ordine cancellato.
- Ricarica avrà lo stesso volume di borsellino.
- Racchiude avrà lo stesso valore dei codici sconto.
- Effettua avrà lo stesso volume degli ordini.
- Per l'associazione contiene abbiamo ipotizzato che in un ordine ci siano in media 3 piatti, quindi moltiplicando per il numero di ordini otteniamo, 250 milioni * 3 = 750 milioni.
- Per l'associazione propone abbiamo pensato che i ristoranti offrissero 50 piatti ciascuno, cioè 200 mila * 50 = 10 milioni.
- Stesura avrà lo stesso volume delle recensioni.
- Per l'associazione scrive abbiamo ipotizzato 4 messaggi per $\frac{3}{4}$ delle 35 milioni di chat

ovvero $4 * 26$ milioni 250 mila = 105 milioni.

- Replica è $\frac{1}{3}$ delle repliche del ristorante all'utente che scrive (associazione), ovvero 34 milioni e mezzo.
- Risponde è $\frac{2}{3}$ delle risposte del rider all'utente che scrive (associazione), ovvero 70 milioni e mezzo.
- Apertura avrà lo stesso volume di consegna non effettuata.
- Avvia avrà lo stesso volume di consegna in ritardo, dato che è una chat che verrà aperta soltanto con questa determinata condizione.
- Produce avrà lo stesso volume di codici sconto.
- Affida avrà lo stesso volume di ordine accettato.
- Abbiamo considerato che un reclamo possa avvenire un 2% delle volte sul totale delle consegne effettuate, quindi il suo volume è 4 000 000.

2.2 Tavola delle operazioni

N°	Operazione	Tipo	Frequenza
1	Effettua un calcolo per aggiornare la classifica dei rider più veloci.	B	1 / mese
2	Effettuare una classifica dei cibi più popolari.	B	1 / mese
3	Effettuare una classifica dei ristoranti con più recensioni positive.	B	1 / mese
4	Calcolare la classifica dei clienti che hanno speso di più.	B	1 / mese
5	Controllare le condizioni di qualifica di top partner e aggiornare lo status dei partner	B	1 / giorno
6	Aggiornare la valutazione media dei ristoranti	B	1 / giorno
7	Controllare la condizione di utente premium e aggiornare lo status degli utenti.	B	1 / giorno
8	Aggiornare le prestazioni di rider	B	1 / giorno
9	Calcolare la distanza rider-ristorante-cliente	B	9000 / giorno
10	Ristorante cancella ordine	I	250 / giorno

11	Ordine effettuato	I	10000 / giorno
12	Scrittura di una recensione	I	1200 / giorno
13	Aggiornamento del menù e della categoria dei piatti	I	500 / giorno

Motivazioni:

- Le operazioni (1,2,3,4) vengono richieste dal testo
- L'operazione (5) riguarda le seguenti condizioni: numero minimo di ordini, valutazione media, percentuale di ordini annullati, percentuale di ordini con reclami
- L'operazione (6) aggiorna la valutazione dei ristoranti quotidianamente.
- L'operazione (7) si assicura che se l'abbonamento viene pagato lo stato dell'utente sia quello premium.
- L'operazione (8) aggiorna quotidianamente le statistiche relative ai rider.
- L'operazione (9) viene calcolata ad ogni ordine per trovare il rider più vicino.
- L'operazione (10) si occupa degli ordini che vengono annullati quotidianamente dai ristoranti, abbiamo supposto che siano il 25% degli ordini totali annullati ogni giorno.
- Un'altra operazione che secondo noi è rilevante è la richiesta di un nuovo ordine e che supponiamo che ne vengano effettuati mediamente 10 000 al giorno (11).
- Un'operazione anche importante è la redazione di una recensione che noi supponiamo siano mediamente 1 200 al giorno in quanto non tutti gli utenti scriveranno la recensione (12).
- L'operazione (13) comprende le modifiche da parte dei ristoranti al proprio menù che può comprendere aggiunta o l'eliminazione di nuovi piatti o la modifica della categoria.

2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

Le ridondanze che abbiamo individuato sono:

- **Numero cancellazioni** (RISTORANTE): attributo derivabile da altre entità.
- **Numero reclami** (RISTORANTE): attributo derivabile da altre entità.
- **Numero stelle** (RISTORANTE): attributo derivabile da altre entità.
- **Valutazione** (RIDER): attributo derivabile da altre entità.
- **Costo totale** (ORDINE): attributo derivabile da un conteggio in altre entità.

Svolgiamo l'analisi delle ridondanze per la ridondanza "Numero cancellazioni".

Riportiamo di seguito due operazioni che riguardano l'entità RISTORANTE (entità che ha come attributo "Numero cancellazioni").

Operazione	Frequenza
Ristorante cancella ordine.	250 / giorno
Controllare le condizioni di qualifica di top partner e aggiornare lo status dei partner.	1 / giorno

Eseguiamo l'analisi delle ridondanze per entrambe le operazioni.

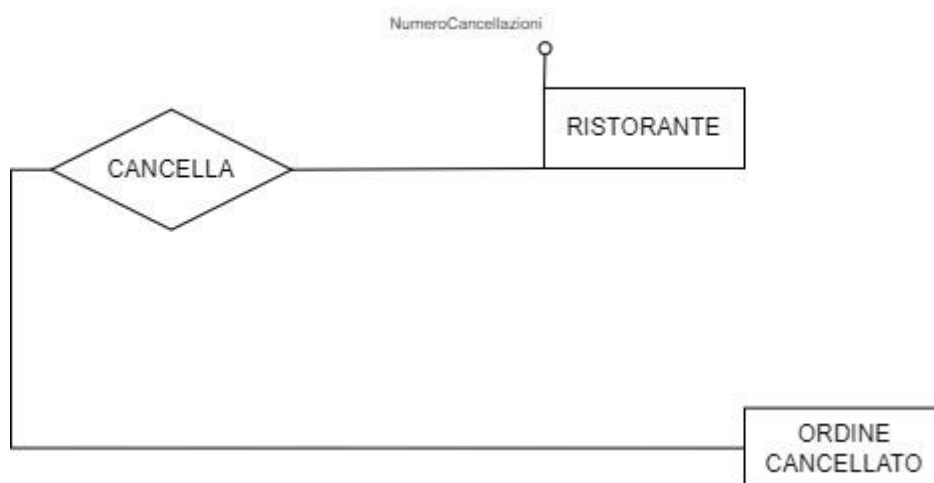
Operazione 1: Ristorante cancella ordine.

- Analisi con ridondanza

1) Tavola dei volumi

Concetto	Tipo	Volume
Ristorante	E	200 000
Cancella	R	8 250 000
Ordine Cancellato	E	25 000 000

2) Schema di navigazione



3) Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Ristorante	Entità	1	S
Cancella	Associazione	1	S
Ordine Cancellato	Entità	1	L

1. Accesso in scrittura a ristorante.

2. Accesso in scrittura a cancella.
3. Accesso in lettura a ordine cancellato.

Totale accessi operazione: 2 accessi in scrittura, 1 in lettura

Totale accessi per giorno: $2 * 250 = 500$ in scrittura,

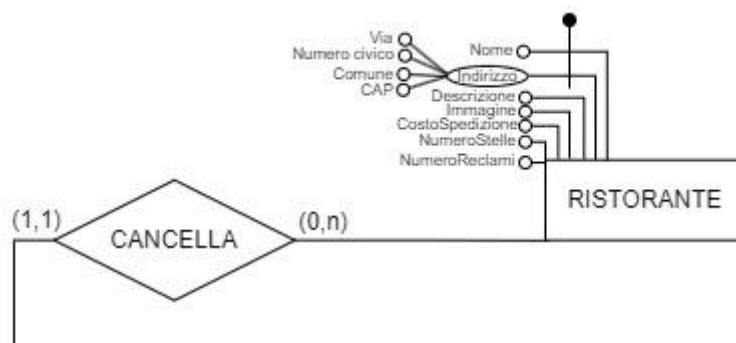
$1 * 1\,000$ (totale ordini cancellati in un giorno) = 1 000 in lettura

- **Analisi senza ridondanza**

- 1) Tavola dei volumi

Concetto	Tipo	Volume
Ristorante	E	200 000
Cancella	A	2 250 000

- 2) Schema di navigazione



- 3) Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Ristorante	E	1	S
Cancella	A	1	S

- 4) Accesso in scrittura in ristorante.
- 5) Accesso in scrittura in cancella.

Totale accessi operazione: 2 accessi in scrittura

Totale accessi per giorno: $2 * 250 = 500$ in scrittura

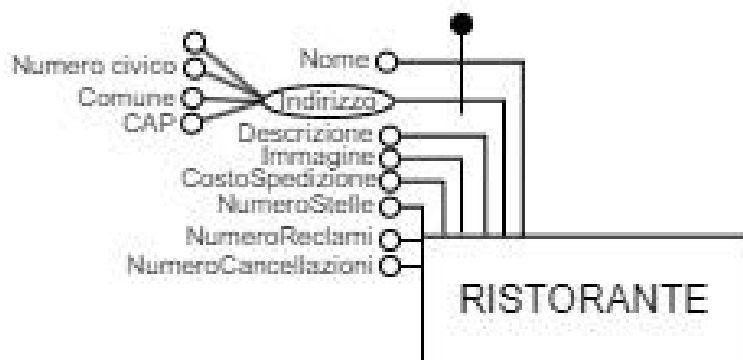
Operazione 2: Controllare le condizioni di qualifica di top partner e aggiornare lo status dei partner.

- **Analisi con ridondanza**

- 1) Tavola dei volumi

Concetto	Tipo	Volume
Ristorante	E	200 000

- 2) Schema di navigazione



- 3) Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Ristorante	Entità	1	L

1. Accesso in lettura in ristorante per leggere i dati del ristorante incluso il numero cancellazioni

Totale accessi operazione: 1 accesso in lettura

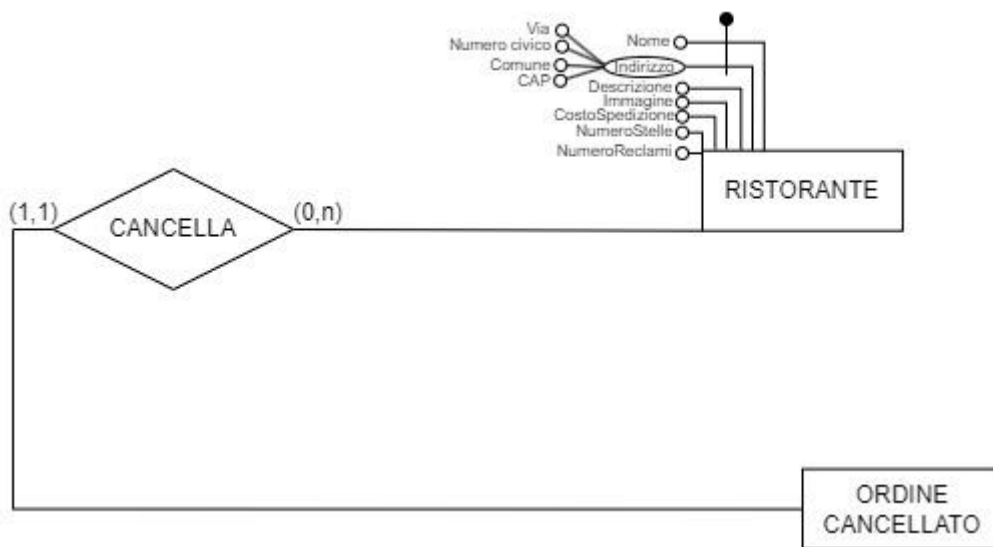
Totale accessi per giorno: $1 * 1 = 1$ in lettura

- **Analisi senza ridondanza**

- 1) Tavola dei volumi

Concetto	Tipo	Volume
Ristorante	E	200 000
Cancella	R	8 250 000
Ordine Cancellato	E	25 000 000

2) Schema di navigazione



3) Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Ristorante	Entità	1	L
Cancella	Associazione	1	L
Ordine Cancellato	Entità	1	L

1. Accesso in lettura a ristorante
2. Accesso in lettura a cancella
3. Accesso in lettura ad ordine cancellato

Totale accessi operazione: 3 accessi in lettura

Totale accessi per giorno: $3 * 1 = 3$ in lettura

Analisi tempo

- Con ridondanza:

- Operazione 1:
500 accessi in scrittura (valgono come 1 000 accessi) e 1 000 accessi in lettura.
- Operazione 2:
1 accessi in lettura.

Totale di 2 001 accessi al giorno.

- Senza ridondanza:
 - Operazione 1:
500 accessi in scrittura (valgono come 1 000 accessi).
 - Operazione 2:
3 accessi in lettura.

Totale di 1 003 accessi al giorno.

Analisi spazio

- Con ridondanza:
Ipotesi: si utilizzano 4 byte per memorizzare l'attributo numero cancellazioni.
Spazio totale necessario: $4 * 200\,000$ (Ristorante) = 800 000 byte
- Senza ridondanza:
Non memorizziamo l'attributo costo totale, per cui non necessita di spazio aggiuntivo.

Con ridondanza	Senza ridondanza
2 001 accessi al giorno	1 003 accessi al giorno
800 000 byte di spazio aggiuntivo	0 byte di spazio aggiuntivo
6 Megabit di spazio aggiuntivo	0 Megabit di spazio aggiuntivo

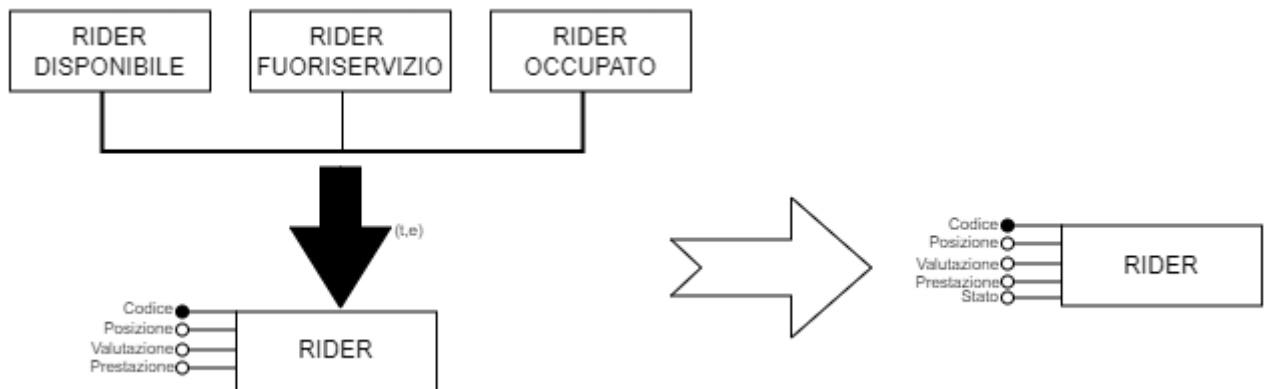
In questo caso scegliamo di togliere la ridondanza perché nonostante utilizza una dimensione di dati esigua rispetto all'intera base dati del servizio (circa 6 Megabit di spazio contro gli zero della versione senza ridondanza) effettua circa 1 000 accessi in più.

2.3.2 Eliminazione delle generalizzazioni

- Entità padre: RIDER

Entità figlie: RIDER DISPONIBILE, RIDER FUORI SERVIZIO, RIDER OCCUPATO

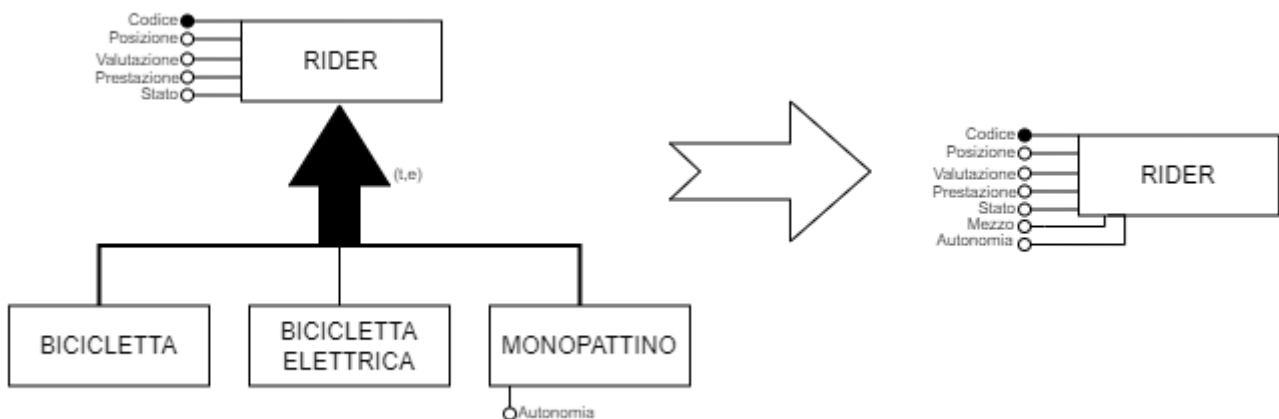
Primo tipo: accorpamento delle entità figlie della generalizzazione nell'entità genitore.



- Entità padre: RIDER

Entità figlie: BICICLETTA , BICICLETTA ELETTRICA , MONOPATTINO

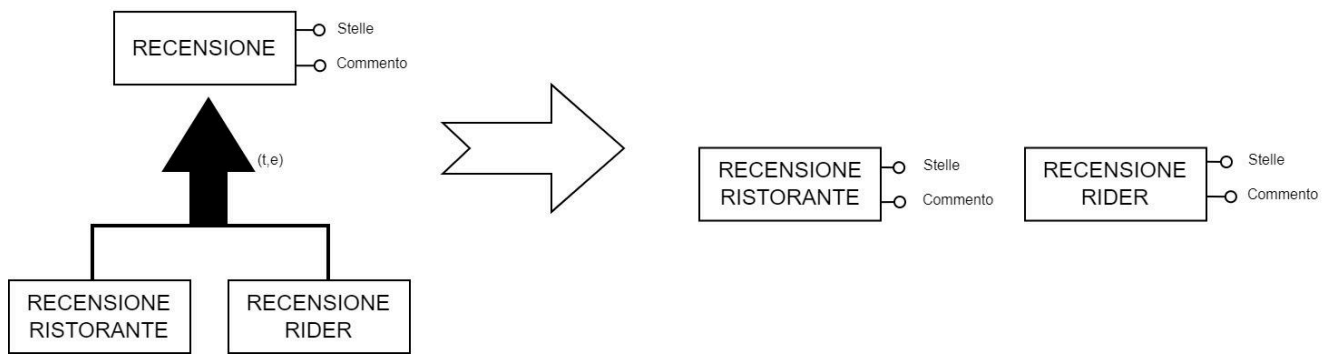
Primo tipo: accorpamento delle entità figlie della generalizzazione nell'entità genitore.



- Entità padre: RECENSIONE AGGIUNGERE ID

Entità figlie: RECENSIONE RIDER , RECENSIONE RISTORANTE

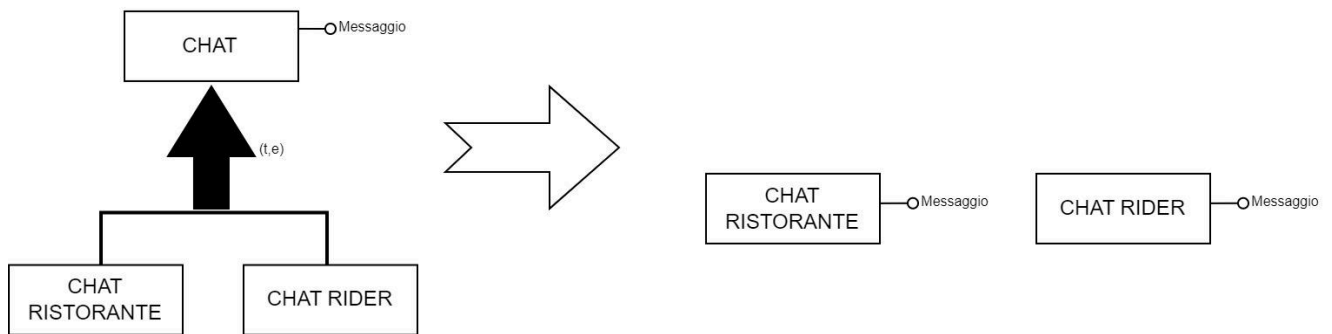
Secondo tipo: accorpamento dell'entità genitore nelle entità figlie



- Entità padre: CHAT AGGIUNGERE ID E ATTRIBUTO

Entità figlia: CHAT RISTORANTE, CHAT RIDER

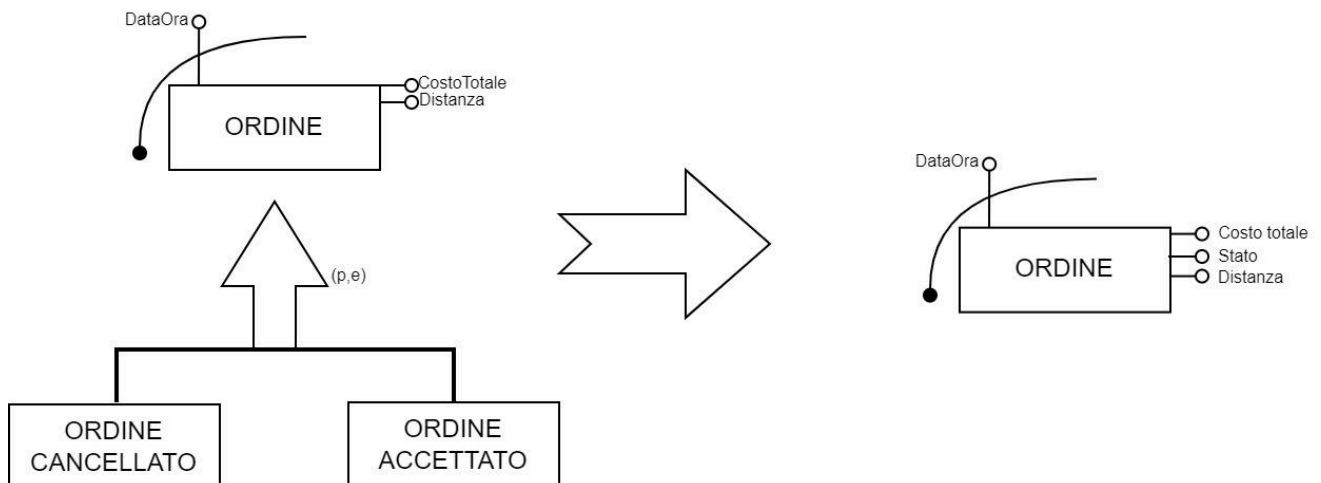
Secondo tipo: accorpamento dell'entità genitore nelle entità figlie



- Entità padre: ORDINE

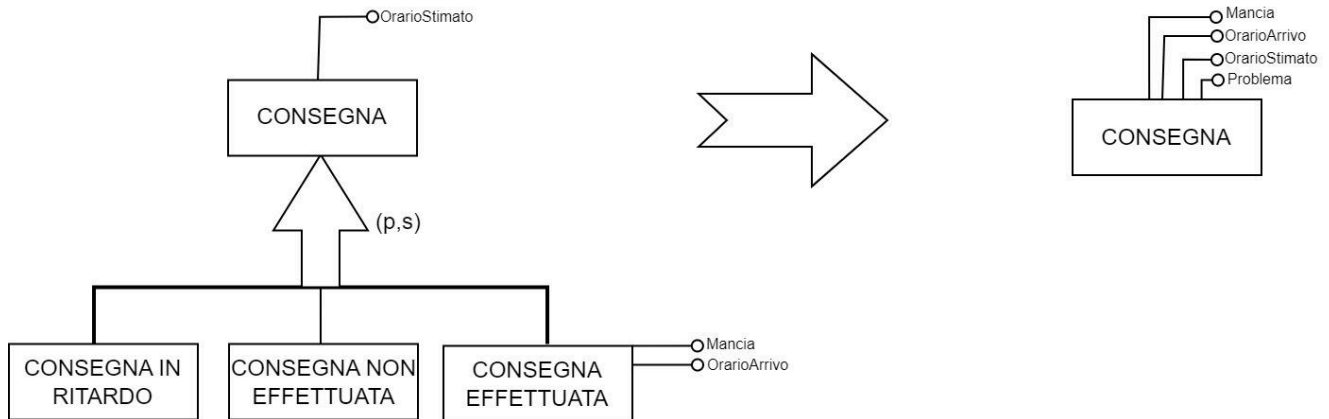
Entità figlia: ORDINE ACCETTATO, ORDINE CANCELLATO

Primo tipo: accorpamento delle entità figlie della generalizzazione nell'entità genitore.



- Entità padre: CONSEGNA

Entità figlie: CONSEGNA IN RITARDO, CONSEGNA NON EFFETTUATA, CONSEGNA EFFETTUATA
Primo tipo: accorpamento delle entità figlie della generalizzazione nell'entità genitore.

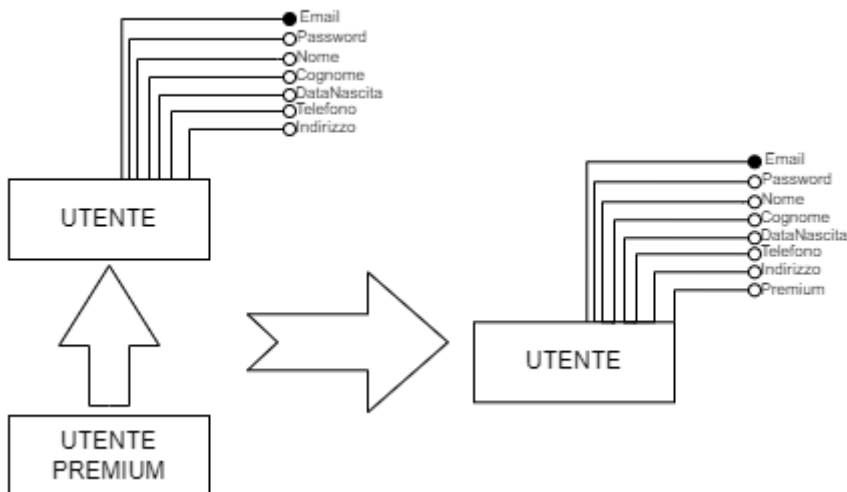


- Entità padre: UTENTE

Entità figlie: UTENTE

PREMIUM

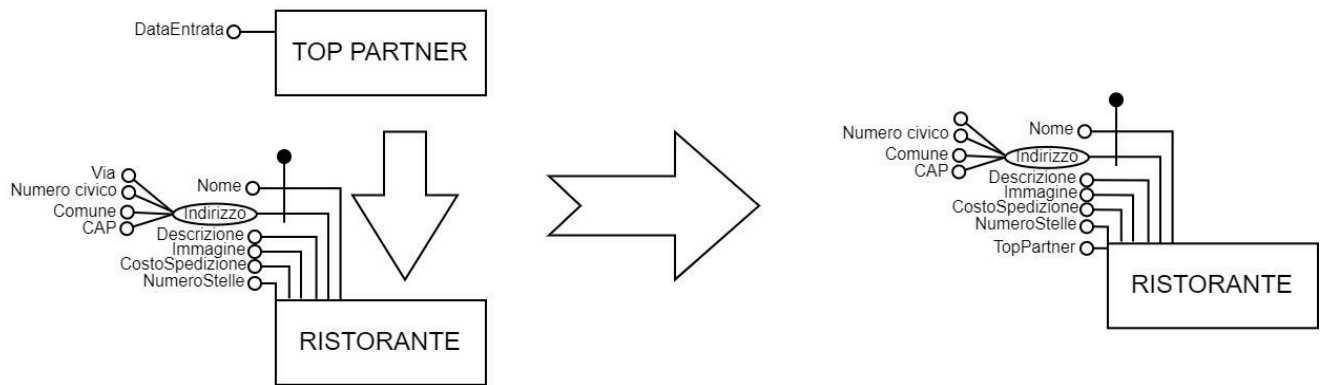
Primo tipo: accorpamento delle entità figlie della generalizzazione nell'entità genitore.



- Entità padre: RISTORANTE

Entità figlie: TOP PARTNER

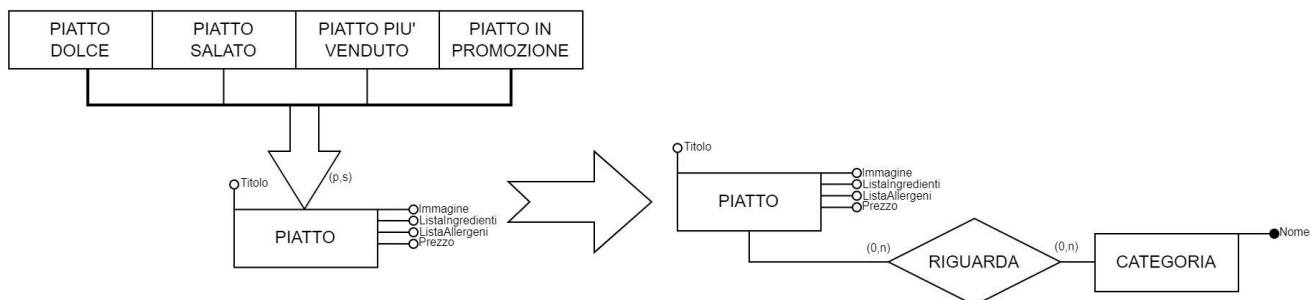
Primo tipo: accorpamento delle entità figlie della generalizzazione nell'entità genitore.



- Entità padre: PIATTO

Entità figlie: PIATTO SALATO, PIATTO DOLCE, PIATTO PIU' VENDUTO, PIATTO IN PROMOZIONE

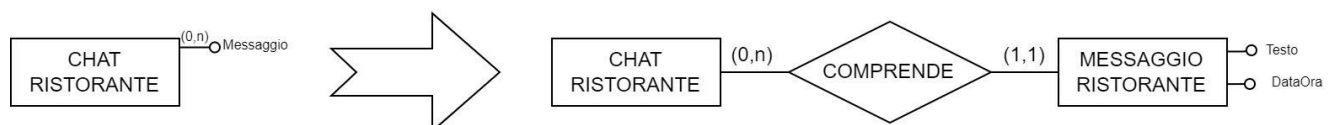
Terzo tipo: sostituzione della generalizzazione con relazioni.



2.3.3 Eventuale partizionamento/accorpamento di entità e associazioni

Partizionamento dell'entità multivalore: AGGIUNGERE CARDINALITA'

- Messaggio in CHAT RISTORANTE



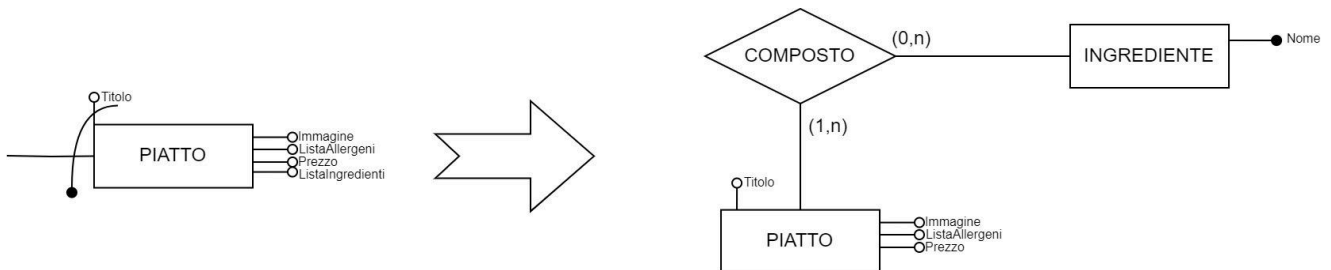
- Messaggio in CHAT RIDER



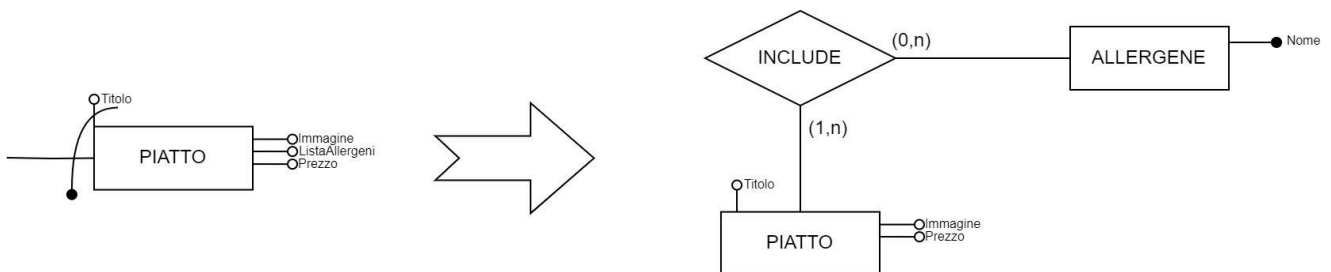
2.3.4 Eventuale eliminazione degli attributi composti e degli attributi multivalore.

Eliminazione degli attributi multivalore:

- Lista ingredienti in PIATTI

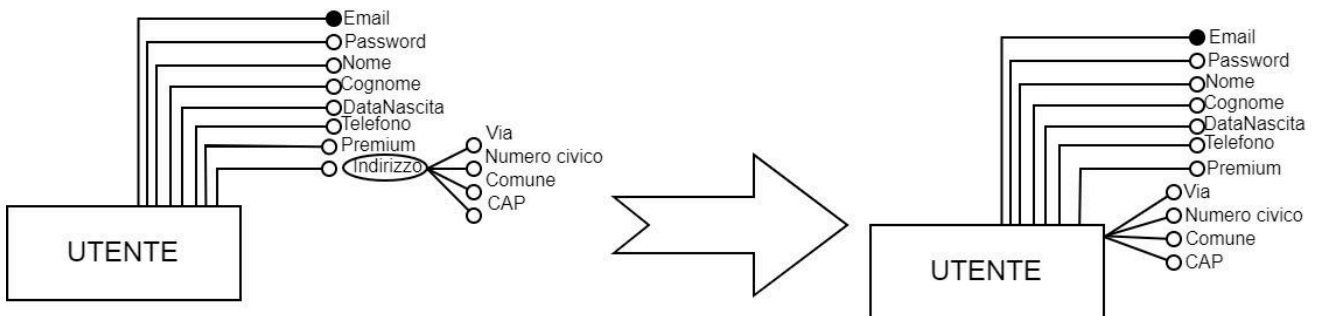


- Lista allergeni in PIATTI

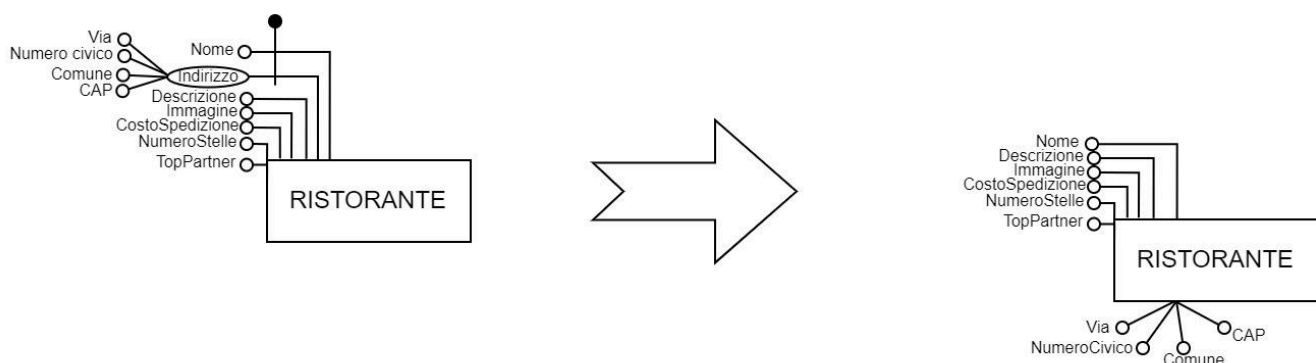


Eliminazione degli attributi composti:

- Indirizzo in UTENTE



- Indirizzo in RISTORANTE

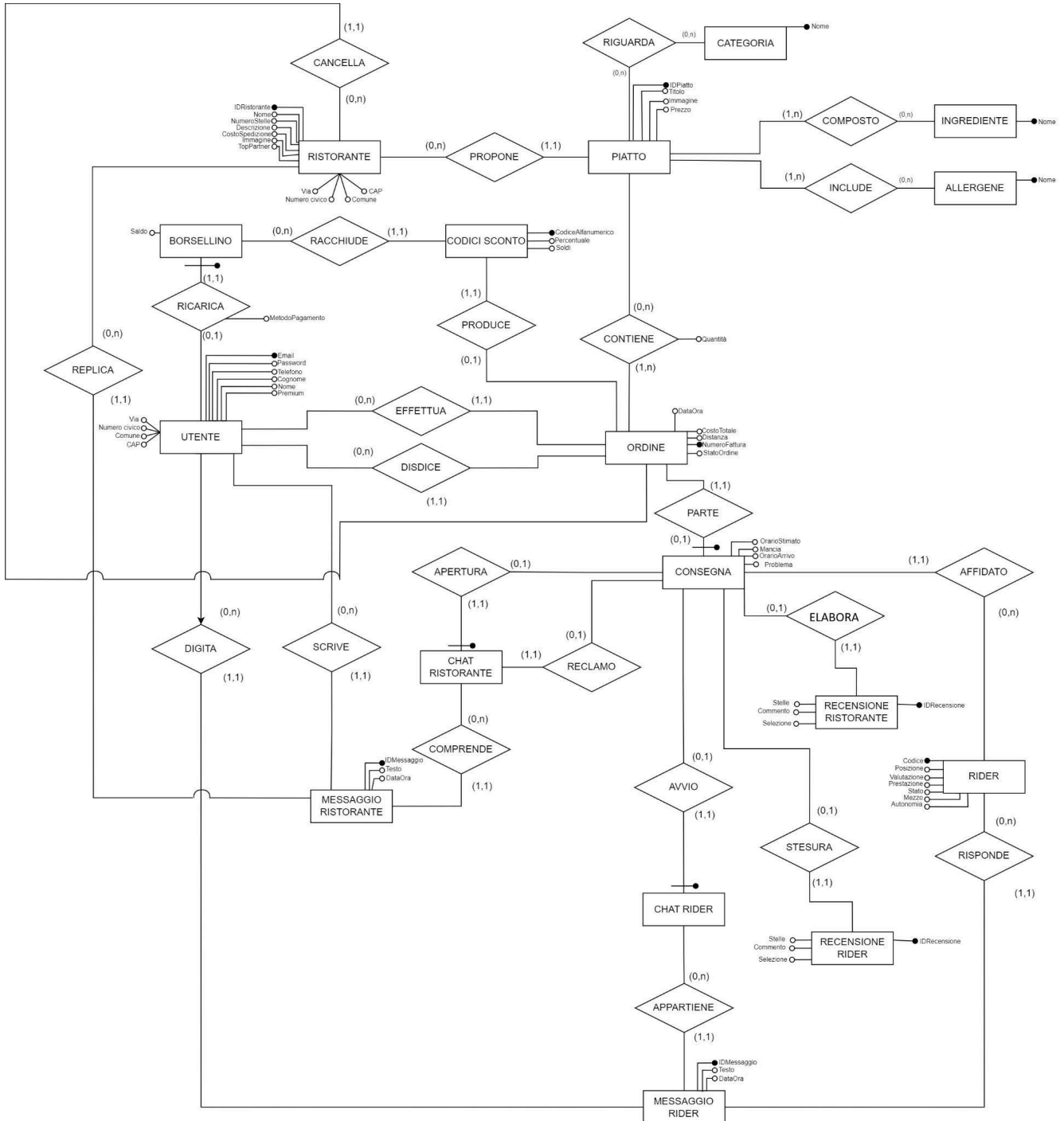


2.3.5 Eventuale scelta degli identificatori principali (con motivazione delle scelte effettuate).

- In UTENTE l'identificatore è e-mail. Supponiamo quindi che ogni utente debba avere un e-mail diversa per registrarsi al servizio.
- In RISTORANTE introduciamo l'identificatore id ristorante (intero autoincrementante) poiché altrimenti le entità che utilizzano questo attributo come chiave si troverebbero un identificatore poco efficiente (una stringa troppo lunga).
- In ORDINE introduciamo l'identificatore NumeroFattura (intero autoincrementante) poiché altrimenti le entità che utilizzano questo attributo come chiave si troverebbero un identificatore poco efficiente (una stringa troppo lunga).
- In CONSEGNA l'identificatore esterno è ORDINE, che referencia NumeroFattura di ORDINE.
- In PIATTO introduciamo l'identificatore id piatto (intero autoincrementante) poiché altrimenti le entità che utilizzano questo attributo come chiave si troverebbero un identificatore poco efficiente (una stringa troppo lunga).
- In RIDER l'identificatore rimane codice.
- In RECENSIONE RIDER introduciamo l'identificatore id recensione (intero autoincrementante) per semplicità.
- In RECENSIONE RISTORANTE introduciamo l'identificatore id recensione (intero autoincrementante) per semplicità.
- In CHAT RISTORANTE introduciamo l'identificatore id chat (intero autoincrementante) per semplicità.
- In CHAT RIDER introduciamo l'identificatore id chat (intero autoincrementante) per semplicità.
- In MESSAGGIO RIDER introduciamo l'identificatore id messaggio (intero autoincrementante).
- In MESSAGGIO RISTORANTE introduciamo l'identificatore id messaggio (intero autoincrementante).
- In INGREDIENTE l'identificatore è Nome.
- In ALLERGENE l'identificatore è Nome.
- in CATEGORIA l'identificatore è Nome.
- In CODICI SCONTO l'identificatore rimane CodiceAlfanumerico.
- In BORSELLINO l'identificatore è esterno con l'entità padre UTENTE.

2.4 Schema E-R ristrutturato + business rules

2.4.1 Schema E-R ristrutturato



2.4.2 Dizionario dei dati (entità)

Entità	Descrizione	Attributi	Identificatore
Utente	Utente del servizio	Email, password, Nome, Cognome, Telefono, Premium*, Via, Numero civico, Comune, CAP	Email
Ristorante	Ristorante del servizio	Nome, Descrizione, CostoSpedizione, Immagine, NumeroStelle, IDRistorante, Via, Numero civico, Comune, CAP, TopPartner*	IDRistorante
Rider	Rider del servizio	Codice, Posizione, Stato, Valutazione, Prestazione, Mezzo, Autonomia	Codice
Ordine	Insieme dei piatti scelti dall'utente	DataOra, CostoTotale, Distanza, NumeroFattura, StatoOrdine	NumeroFattura
Consegna	Trasferimento di quanto ordinato	OrarioStimato, Mancias, OrarioArrivo, Problema	Ordine
Piatto	Piatto proposto dal ristorante	Titolo, Immagine, Prezzo, IDPiatto	IDPiatto
Categoria	Categoria del piatto	Nome	Nome
Ingrediente	Ingrediente del piatto	Nome	Nome
Allergene	Allergene del piatto	Nome	Nome
Borsellino	Salvadanaio digitale dell'utente	Saldo	Utente
Recensione Ristorante	Recensione dell'utente al ristorante	Stelle, Commento, IDRecensione	IDRecensione

Recensione Rider	Recensione dell'utente al rider	Stelle, Commento, IDRecensione	IDRecensione
Chat Rider	Messaggi tra utente e rider	IDChat	IDChat
Chat Ristorante	Messaggi tra utente e ristorante	IDChat	IDChat
Messaggio Rider	Testo scritto da utente o rider	IDMessaggio, Testo, DataOra	IDMessaggio
Messaggio Ristorante	Testo scritto da utente o ristorante	IDMessaggio, Testo, DataOra	IDMessaggio
Codici Sconto	Codice che permette di ottenere uno sconto	CodiceAlfanumerico, Percentuale, Soldi	CodiceAlfanumerico

2.4.3 Dizionario dei dati (relazioni)

Relazioni	Descrizione	Componenti	Attributi
Disdice	Disdetta dell'ordine da parte dell'utente	Utente (0,n), Ordine (1,1)	
Cancella	Cancellazione dell'ordine da parte del ristorante	Ristorante (0,n), Ordine (1,1)	
Ricarica	Ricarica del borsellino	Utente (0,1), Borsellino (1,1)	Metodo di pagamento
Racchiude	Codici sconto racchiusi nel borsellino	Borsellino (0,n), Codici sconto (1,1)	
Effettua	Ordine effettuato da un utente	Utente (0,n), Ordine (1,1)	
Contiene	Piatto contenuto nell'ordine	Ordine (1,n), Piatto (0,n)	Quantità
Propone	Piatti proposti dal ristorante	Ristorante (0,n), Piatto (1,1)	
Parte	Consegna eseguita dal rider	Ordine (1,1), Consegna (0,1)	
Stesura	Stesura di una recensione al rider	Consegna (0,1), Recensione Rider (1,1)	

Elabora	Elaborazione della recensione al ristorante	Consegna(0,1), Recensione Ristorante (1,1)	
Scrive	Scrittura di un messaggio tra utente e ristorante	Utente (0,n), MessaggioRistorante (1,1)	
Digita	Scrittura di un messaggio tra utente e rider	Utente (0,n), MessaggioRider (1,1)	
Replica	Replica ad un messaggio dal ristorante	Ristorante (0,n), MessaggioRistorante (1,1)	
Risponde	Risposta ad un messaggio dal rider	Rider (0,n), MessaggioRider (1,1)	
Apertura	Apertura della chat del rider	Consegna (0,1), ChatRistorante (1,1)	
Avvia	Avviamento della chat del ristorante	Consegna (0,1), ChatRider (1,1)	
Produce	Codici sconto prodotti dall'ordine	Ordine (0,1), Codici sconto (1,1)	
Riguarda	Categoria riguardante un piatto	Piatto (0,n), Categoria (0,n)	
Composto	Ingrediente compone un piatto	Piatto (1,n), Ingrediente (0,n)	
Include	Allergene incluso in un piatto	Piatto (1,n), Allergene (0,n)	
Reclamo	Ordine consegnato reclamato nella chat del ristorante	Consegna(0,1) ChatRistorante (1,1)	
Appartiene	Messaggio che appartiene alla chat ristorante	ChatRistorante (0,n), MessaggioRistorante (1,1)	
Comprende	Messaggio compreso nella chat rider	ChatRider (0,n), MessaggioRider (1,1)	

2.4.4 Vincoli d'integrità

1. Ogni ristorante **deve** avere una valutazione quinquaria aggiornata ogni lunedì sulla base della media delle proprie recensioni.
2. Il ristorante Top Partner **deve** avere consegnato correttamente almeno 20 ordini.
3. Il ristorante Top Partner **deve** avere una valutazione utenti maggiore o uguale a 4.5 stelline su cinque.
4. Il ristorante Top Partner **deve** avere una percentuale massima di ordini annullati dal ristorante dell'1.5% e una percentuale massima di ordini con reclami da parte degli utenti del 2.5%.
5. La recensione **deve** diventare visibile quando sia host che ospite hanno fatto la recensione
6. Gli ordini **devono** poter essere annullati sia dall'utente che dal ristorante fino al momento della consegna dell'ordine al rider.
7. L'utente può recensire sia il rider che il ristorante con una valutazione che **deve** essere compresa tra 1 e 5.
8. L'attributo Premium **deve** essere la data di inizio dell'abbonamento premium dell'utente, se questo ha sottoscritto l'abbonamento.
9. L'attributo TopPartner **deve** essere TRUE se il ristorante soddisfa tutti i requisiti.

2.4.5 Derivazioni

1. La valutazione del ristorante **si ottiene** facendo la media tra le valutazioni ricevute dagli utenti che hanno effettuato un ordine.
2. La valutazione del rider **si ottiene** facendo la media tra le valutazioni ricevute dagli utenti che hanno ricevuto l'ordine.
3. La percentuale di ordini annullati **si ottiene** dal rapporto degli ordini annullati con il totale degli ordini effettuati.
4. La percentuale di ordini con reclami **si ottiene** dal rapporto di ordini che hanno riscontrato problemi con il numero di ordini consegnati.
5. Il numero di recensioni del ristorante **si ottiene** conteggiando le recensioni riguardanti gli ordini consegnati che fanno riferimento a quel determinato ordine.
6. Il costo totale **si ottiene** sommando i costi dei piatti ordinati con il prezzo della consegna.
7. La prestazione del rider **si ottiene** sottraendo il numero di ordini consegnati con i minuti di ritardo totali e sottraendo ulteriormente il numero di ordini non consegnati * 2, la moltiplicazione per 2 è un modo per dare un peso maggiore agli ordini non consegnati rispetto a quelli consegnati.

2.5 Schema relazionale + vincoli di integrità referenziale

2.5.1 Schema relazionale

UTENTE (Email, Password, Cognome, Nome, Telefono, DataNascita, Premium*, Via, NumeroCivico, Comune, CAP)

RISTORANTE (IDRistorante, Nome, Descrizione, CostoSpedizione, Immagine, NumeroStelle, Via, NumeroCivico, Comune, CAP, TopPartner*)

RIDER (Codice, Posizione, Stato, Valutazione, Prestazione, Mezzo, Autonomia)

ORDINE (NumeroFattura, DataOra, CostoTotale, Distanza, Cancellazione, Utente)

CONSEGNA (Ordine, OrarioStimato, Mancia, OrarioArrivo, Problema, Rider)

PIATTO (IDPiatto, Titolo, Immagine, Prezzo, Ristorante)

CATEGORIA (Nome)

INGREDIENTE(Nome)

ALLERGENE (Nome)

BORSELLINO(Utente, Saldo)

RICARICA(IDRicarica, IDBorsellino, MetodoDiPagamento, Movimento)

RECENSIONE RISTORANTE(IDRecensione, Commento, Stelle, Consegna)

RECENSIONE RIDER(IDRecensione, Commento, Stelle, Consegna)

CHAT RIDER(IDChat, Consegna)

CHAT RISTORANTE(IDChat, Consegna)

MESSAGGIO RIDER(IDMessaggio, Testo, DataOra, Rider, Utente, ChatRider)

MESSAGGIO RISTORANTE(IDMessaggio, Testo, DataOra, Ristorante, Utente, ChatRistorante)

CODICE SCONTO(CodiceAlfanumerico, Percentuale, Soldi, Borsellino, Ordine)

2.5.2 Vincoli di integrità referenziale

Con vincoli d'integrità referenziale tra:

- L'attributo utente in ORDINE e la chiave di UTENTE.
- L'attributo utente in BORSELLINO e la chiave di UTENTE.
- L'attributo borsellino in RICARICA e la chiave di BORSELLINO.

- L'attributo borsellino in CODICI SCONTO e la chiave di BORSELLINO.
- L'attributo ristorante in PIATTO e la chiave di RISTORANTE.
- L'attributo piatto in ORDINE-PIATTO e la chiave di PIATTO.
- L'attributo piatto in INGREDIENTE-PIATTO e la chiave di PIATTO.
- L'attributo piatto in ALLERGENE-PIATTO e la chiave di PIATTO.
- L'attributo piatto in CATEGORIA-PIATTO e la chiave di PIATTO.
- L'attributo consegna in RECENSIONE-RIDER e la chiave di CONSEGNA.
- L'attributo consegna in RECENSIONE-RISTORANTE e la chiave di CONSEGNA.
- L'attributo ingrediente in INGREDIENTE-PIATTO e la chiave di INGREDIENTE.
- L'attributo allergene in ALLERGENE-PIATTO e la chiave di ALLERGENE.
- L'attributo categoria in CATEGORIA-PIATTO e la chiave di CATEGORIA.
- L'attributo ordine in ORDINE-PIATTO e la chiave di ORDINE.
- L'attributo ordine in CODICI SCONTO e la chiave di ORDINE.
- L'attributo ordine in CONSEGNA e la chiave di ORDINE.
- L'attributo ordine in CHAT RIDER e la chiave di CONSEGNA.
- L'attributo ordine in CHAT RISTORANTE e la chiave di CONSEGNA.
- L'attributo rider in CONSEGNA e la chiave di RIDER.
- L'attributo chat rider in MESSAGGIO RIDER e la chiave di CHAT RIDER.
- L'attributo utente in MESSAGGIO RIDER e la chiave di UTENTE.
- L'attributo rider in MESSAGGIO RIDER e la chiave di RIDER.
- L'attributo chat ristorante in MESSAGGIO RISTORANTE e la chiave di CHAT RISTORANTE.
- L'attributo utente in MESSAGGIO RISTORANTE e la chiave di UTENTE.
- L'attributo ristorante in MESSAGGIO RISTORANTE e la chiave di RISTORANTE.

3. Implementazione

3.1 DDL di creazione del database

```
CREATE TABLE Utente(
    email varchar (50) CHECK (email LIKE '%@%.%'),
    password varchar (50) NOT NULL,
    telefono varchar (30) NOT NULL,
    dataNascita date NOT NULL,
    nome varchar (20) NOT NULL,
    cognome varchar (20) NOT NULL,
    premium boolean NOT NULL,
    via varchar (40) NOT NULL,
    CAP integer NOT NULL CHECK (CAP >= 00000 and CAP <= 99999),
    numeroCivico SMALLINT NOT NULL CHECK (numeroCivico>0),
    comune varchar (40) NOT NULL,
    CONSTRAINT Utente_pk PRIMARY KEY (email)
);
```



```
CREATE TABLE Ristorante(
    id SERIAL,
    nome varchar (50) NOT NULL,
    descrizione text NOT NULL,
    costoSpedizione integer NOT NULL,
    immagine bytea,
    numeroStelle numeric(1) CHECK(numeroStelle<=5 AND numerostelle>=1) NOT
    via varchar (40) NOT NULL,
    CAP integer NOT NULL CHECK (CAP >= 00000 and CAP <= 99999),
    numeroCivico SMALLINT NOT NULL CHECK (numeroCivico>0),
    comune varchar (40) NOT NULL,
    topPartner boolean NOT NULL,
    CONSTRAINT Ristorante_PK PRIMARY KEY (id)
);
```

```
CREATE TABLE Rider(
    codice SERIAL,
    posizione varchar (50) NOT NULL,
    valutazione numeric(1) CHECK(valutazione<=5 AND valutazione>=1),
    prestazione integer NOT NULL,
    stato varchar (20) NOT NULL
    CHECK(
    stato LIKE 'fuori servizio' OR
    stato LIKE 'disponibile' OR
    stato LIKE 'occupato'
    ),
    mezzo varchar (20) NOT NULL
    CHECK(
    mezzo LIKE 'bici' OR
    mezzo LIKE 'bici elettrica' OR
    mezzo LIKE 'monopattino'
    ),
    autonomia integer,
    CONSTRAINT Rider_PK PRIMARY KEY (codice)
);
```

```
CREATE TABLE Ricarica(
    id SERIAL,
    metodoPagamento varchar (16) NOT NULL
    CHECK(
    metodoPagamento LIKE 'carta di credito' OR
    metodoPagamento LIKE 'paypal' OR
    metodoPagamento LIKE 'bancomat' OR
    metodoPagamento LIKE 'satispay'
    ),
    movimento integer NOT NULL,
    borsellino varchar (50) NOT NULL,
    FOREIGN KEY(borsellino) references borsellino(utente)
    ON UPDATE CASCADE,
    CONSTRAINT Ricarica_PK PRIMARY KEY (id)
);
```

```
CREATE TABLE Piatto(
    id SERIAL,
    titolo varchar (50) NOT NULL,
    immagine bytea,
    prezzo float NOT NULL,
    ristorante integer NOT NULL,

    FOREIGN KEY(ristorante) references Ristorante(id),
    CONSTRAINT Piatto_PK PRIMARY KEY (id)
);
```

```
CREATE TABLE Borsellino(
    utente varchar (50) NOT NULL,
    saldo integer NOT NULL,
    FOREIGN KEY(utente) references Utente(email)
    ON UPDATE CASCADE,
    CONSTRAINT Borsellino_PK PRIMARY KEY (utente)
);
```

```

CREATE TABLE Ingrediente(
    nome varchar (50) NOT NULL,
    CONSTRAINT Ingrediente_PK PRIMARY KEY (nome)
);|

CREATE TABLE Allergene(
    nome varchar (50) NOT NULL,
    CONSTRAINT Allergene_PK PRIMARY KEY (nome)
);|

CREATE TABLE Categoria(
    nome varchar (50) NOT NULL,
    CONSTRAINT Categoria_PK PRIMARY KEY (nome)
);

CREATE TABLE AllergenePiatto(
    piatto integer NOT NULL,
    Allergene varchar (50) NOT NULL,
    FOREIGN KEY(piatto) references Piatto(id),
    FOREIGN KEY(allergene) references Allergene(nome),
    CONSTRAINT AllergenePiatto_PK PRIMARY KEY (piatto,allergene)
);|

CREATE TABLE IngredientePiatto(
    piatto integer NOT NULL,
    ingrediente varchar (50) NOT NULL,
    FOREIGN KEY(piatto) references Piatto(id),
    FOREIGN KEY(ingrediente) references Ingrediente(nome),
    CONSTRAINT IngredientePiatto_PK PRIMARY KEY (piatto,ingrediente)
);|

CREATE TABLE CategoriaPiatto(
    piatto integer NOT NULL,
    categoria varchar (50) NOT NULL,
    FOREIGN KEY(piatto) references Piatto(id),
    FOREIGN KEY(categoria) references Categoria(nome),
    CONSTRAINT CategoriaPiatto_PK PRIMARY KEY(piatto,categoria)
);

CREATE TABLE OrdinePiatto(
    piatto integer NOT NULL,
    ordine integer NOT NULL,
    quantità integer NOT NULL,
    FOREIGN KEY(piatto) references Piatto(id),
    FOREIGN KEY(ordine) references Ordine(numeroFattura),
    CONSTRAINT OrdinePiatto_PK PRIMARY KEY (piatto,ordine)
);|

CREATE TABLE Ordine(
    numeroFattura SERIAL,
    dataOra timestamp NOT NULL,
    costoTotale float NOT NULL,
    distanza float NOT NULL,
    utente varchar (50) NOT NULL,
    FOREIGN KEY(utente) references Utente(email)
    ON UPDATE CASCADE,
    CONSTRAINT Ordine_PK PRIMARY KEY (numeroFattura)
)

```

```
CREATE TABLE CodiceSconto(
    id SERIAL,
    percentuale integer CHECK(percentuale>0 AND percentuale<=100),
    soldi integer,
    borsellino varchar(50) NOT NULL,
    ordine integer NOT NULL,

    FOREIGN KEY(borsellino) references Borsellino(utente)
    ON UPDATE CASCADE,
    FOREIGN KEY(ordine) references Ordine(numeroFattura),
    CONSTRAINT CodiceSconto_PK PRIMARY KEY (id)
);|
```

```
CREATE TABLE Consegna(
    ordine integer NOT NULL,
    orarioStimato time NOT NULL,
    mancia float,
    orarioArrivo time,
    rider integer NOT NULL,
    problema varchar(30)
    CHECK(
        problema LIKE 'consegna non effettuata' OR
        problema LIKE 'netto ritardo' OR
        problema LIKE 'reclamo'
    ),
    FOREIGN KEY(rider) references Rider(codice),
    FOREIGN KEY(ordine) references Ordine(numeroFattura),
    CONSTRAINT Consegna_PK PRIMARY KEY(ordine)
);
```

```
CREATE TABLE ChatRider(
    id SERIAL,
    consegna integer NOT NULL,
    FOREIGN KEY(consegna) references Consegna(ordine),
    CONSTRAINT ChatRider_PK PRIMARY KEY (id)
);|
```

```
CREATE TABLE ChatRistorante(
    id SERIAL,
    consegna integer NOT NULL,
    FOREIGN KEY(consegna) references Consegna(ordine),
    CONSTRAINT ChatRistorante_PK PRIMARY KEY (id)
);|
```

```
CREATE TABLE MessaggioRistorante(
    id SERIAL,
    testo text NOT NULL,
    DataOra timestamp,

    utente varchar (50) NOT NULL,
    ristorante integer NOT NULL,
    chatRistorante integer NOT NULL,
    FOREIGN KEY(ristorante) references Ristorante(id),
    FOREIGN KEY(chatRistorante) references ChatRistorante(id),
    FOREIGN KEY(utente) references Utente(email)
    ON UPDATE CASCADE,
    CONSTRAINT MessaggioRistorante_PK PRIMARY KEY (id)
);
```

```

CREATE TABLE MessaggioRider(
    id SERIAL,
    testo text NOT NULL,
    dataOra timestamp,

    utente varchar (50) NOT NULL,
    rider integer NOT NULL,
    chatRider integer NOT NULL,
    FOREIGN KEY(rider) references Rider(codice),
    FOREIGN KEY(chatRider) references ChatRider(id),
    FOREIGN KEY(utente) references Utente(email)
    ON UPDATE CASCADE,
    CONSTRAINT MessaggioRider_PK PRIMARY KEY (id)
);

CREATE TABLE RecensioneRider(
    id SERIAL,
    testo text,
    stelle numeric(1) NOT NULL CHECK(stelle<=5 AND stelle>=1),
    consegna integer NOT NULL,
    FOREIGN KEY(consegna) references Consegna(ordine),
    CONSTRAINT RecensioneRider_PK PRIMARY KEY (id)
);

CREATE TABLE RecensioneRistorante(
    id SERIAL,
    testo text ,
    stelle numeric(1) NOT NULL CHECK(stelle<=5 AND stelle>=1),
    consegna integer NOT NULL,
    FOREIGN KEY(consegna) references Consegna(ordine),
    CONSTRAINT RecensioneRistorante_PK PRIMARY KEY (id)
);

```

3.2 DML di popolamento di tutte le tabelle del database

```

INSERT INTO Utente (email, password, telefono, dataNascita, nome, cognome, premium, via, CAP, numeroCivico, comune)
VALUES
('mario.rossi@example.com', 'password1', '1234567890', '1990-05-20', 'Mario', 'Rossi', true, 'Via Roma', 12345, 10, 'Roma'),
('laura.bianchi@example.com', 'password2', '0987654321', '1985-10-15', 'Laura', 'Bianchi', false, 'Via Milano', 54321, 20, 'Milano'),
('luigi.verdi@example.com', 'password3', '1122334455', '2000-03-08', 'Luigi', 'Verdi', true, 'Via Napoli', 67890, 30, 'Napoli'),
('giovanna.russo@example.com', 'password4', '5544332211', '1998-12-01', 'Giovanna', 'Russo', false, 'Via Firenze', 45678, 40, 'Firenze'),
('francesca.ferrari@example.com', 'password5', '6677889900', '1992-07-25', 'Francesca', 'Ferrari', true, 'Via Bologna', 23456, 50, 'Bologna'),
('paolo.ricci@example.com', 'password6', '3344556677', '1987-09-14', 'Paolo', 'Ricci', false, 'Via Venezia', 78901, 60, 'Venezia'),
('alessia.martini@example.com', 'password7', '8899001122', '1995-04-30', 'Alessia', 'Martini', true, 'Via Torino', 56789, 70, 'Torino'),
('marco.moretti@example.com', 'password8', '2211003344', '1983-11-18', 'Marco', 'Moretti', false, 'Via Genova', 34567, 80, 'Genova'),
('giulia.conti@example.com', 'password9', '4433221100', '1979-08-03', 'Giulia', 'Conti', true, 'Via Palermo', 90123, 90, 'Palermo'),
('roberto.gallo@example.com', 'password10', '7788990011', '1993-02-17', 'Roberto', 'Gallo', false, 'Via Catania', 67812, 100, 'Catania');

INSERT INTO Ristorante (nome, descrizione, costoSpedizione, immagine, numeroStelle, via, CAP, numeroCivico, comune, topPartner)
VALUES
('La Pizzeria di Paolo', 'Autentica pizza napoletana preparata con ingredienti freschi e genuini.', 3, NULL, 4.5, 'Via Roma', 20121, 10, 'Milano', true),
('Ristorante da Giovanni', 'Cucina tradizionale toscana con una vasta selezione di vini locali.', 5, NULL, 4, 'Via Garibaldi', 50100, 15, 'Firenze', false),
('Sushi House', 'Specialità giapponesi preparate al momento da chef esperti.', 4, NULL, 4.2, 'Corso Italia', 10100, 27, 'Torino', true),
('Trattoria Maria', 'Piatti tipici della cucina romana, dalla amatriciana alla coda alla vaccinara.', 3, NULL, 4.3, 'Via Veneto', 00100, 5, 'Roma', false),
('Il Gourmet', 'Cucina creativa e innovativa con ingredienti freschi e di stagione.', 6, NULL, 4.7, 'Via dei Mille', 40100, 20, 'Bologna', true),
('Ristorante del Mare', 'Pesce fresco ogni giorno direttamente dai pescatori locali.', 7, NULL, 4.6, 'Lungomare', 70100, 30, 'Napoli', false),
('Steakhouse Grill', 'Bistecche succulente e grigliate perfette, accompagnate da contorni deliziosi.', 8, NULL, 4.4, 'Via Roma', 20121, 12, 'Milano', true),
('Cantina dei Saporì', 'Ampia selezione di vini italiani da abbinare ai piatti della tradizione regionale.', 4, NULL, 4.1, 'Piazza del Popolo', 00186, 8, 'Roma', false),
('Il Pasticciere', 'Dolci fatti in casa, torte decorate e pasticcini per ogni occasione.', 3, NULL, 4.8, 'Via Garibaldi', 50100, 25, 'Firenze', true),
('La Taverna di Luca', 'Atmosfera accogliente e piatti della cucina casalinga con un tocco gourmet.', 5, NULL, 4.3, 'Via Dante', 10100, 18, 'Torino', false);

```

```
INSERT INTO Rider (posizione, valutazione, prestazione, stato, mezzo, autonomia)
```

```
VALUES
```

```
(45.4641-9.1919, 4.5, 20, 'disponibile', 'bici', NULL),  
(45.0705-7.6869, 4.2, 18, 'disponibile', 'monopattino', 30),  
(43.7730-11.2555, 4.8, 22, 'disponibile', 'bici elettrica', NULL),  
(41.9028-12.4964, 4.3, 17, 'occupato', 'bici', NULL),  
(45.0704-7.6869, 4.7, 21, 'disponibile', 'monopattino', 25),  
(41.9062-12.4884, 4.1, 19, 'disponibile', 'bici elettrica', NULL),  
(40.8518-14.2681, 4.6, 23, 'disponibile', 'bici', NULL),  
(43.7228-10.3944, 4.4, 20, 'fuori servizio', 'bici', NULL),  
(43.7228-10.3944, 4.9, 24, 'disponibile', 'bici elettrica', NULL),  
(45.4642-9.1886, 4.0, 16, 'disponibile', 'monopattino', 35);
```

```
INSERT INTO Allergene (nome)
```

```
VALUES
```

```
('Glutine'),  
( 'Lattosio'),  
( 'Arachidi'),  
( 'Crostacei'),  
( 'Soia');
```

```
INSERT INTO Ingrediente (nome)
```

```
VALUES
```

```
('Pomodoro'),  
( 'Cipolla'),  
( 'Aglio'),  
( 'Basilico'),  
( 'Peperoncino'),  
( 'Parmigiano'),  
( 'Mozzarella'),  
( 'Prosciutto'),  
( 'Funghi');
```

```
INSERT INTO Categoria (nome)
```

```
VALUES
```

```
('Dolce'),  
( 'Salato'),  
( 'Più venduto'),  
( 'In promozione');
```

```
INSERT INTO Borsellino (utente, saldo)
```

```
VALUES
```

```
('mario.rossi@example.com', 150),  
( 'laura.bianchi@example.com', 200),  
( 'luigi.verdi@example.com', 75),  
( 'giovanna.russo@example.com', 300),  
( 'francesca.ferrari@example.com', 100),  
( 'paolo.ricci@example.com', 250),  
( 'alessia.martini@example.com', 180),  
( 'marco.moretti@example.com', 220),  
( 'giulia.conti@example.com', 90),  
( 'roberto.gallo@example.com', 210);
```

```
INSERT INTO Ricarica (metodoPagamento, movimento, borsellino)
```

```
VALUES
```

```
('carta di credito', 50, 'mario.rossi@example.com'),
('paypal', 30, 'laura.bianchi@example.com'),
('bancomat', 20, 'luigi.verdi@example.com'),
('satispay', 40, 'giovanna.russo@example.com'),
('carta di credito', 25, 'francesca.ferrari@example.com'),
('paypal', 35, 'paolo.ricci@example.com'),
('bancomat', 45, 'alessia.martini@example.com'),
('satispay', 60, 'marco.moretti@example.com'),
('carta di credito', 55, 'giulia.conti@example.com'),
('paypal', 70, 'roberto.gallo@example.com');|
```

```
INSERT INTO Piatto (titolo, immagine, prezzo, ristorante)
```

```
VALUES
```

```
('Pizza Margherita', NULL, 8.99, 1),
('Spaghetti Carbonara', NULL, 12.50, 2),
('Sushi Misto', NULL, 15.99, 3),
('Tagliata di Manzo', NULL, 18.75, 4),
('Insalata Caprese', NULL, 9.99, 1),
('Risotto ai Frutti di Mare', NULL, 14.50, 2),
('Sashimi di Salmone', NULL, 16.99, 3),
('Cotoletta alla Milanese', NULL, 17.75, 4),
('Tiramisù', NULL, 6.99, 1),
('Gelato alla Vaniglia', NULL, 4.50, 2);
```

```
INSERT INTO Ordine (dataOra, costoTotale, distanza, utente)
```

```
VALUES
```

```
('2024-05-11 12:30:00', 45.99, 5.2, 'mario.rossi@example.com'),
('2024-05-11 19:15:00', 28.50, 3.8, 'laura.bianchi@example.com'),
('2024-05-12 20:00:00', 33.99, 6.5, 'luigi.verdi@example.com'),
('2024-05-12 18:45:00', 52.75, 8.1, 'giovanna.russo@example.com'),
('2024-05-13 13:00:00', 27.99, 4.3, 'francesca.ferrari@example.com'),
('2024-05-13 21:30:00', 45.50, 7.6, 'paolo.ricci@example.com'),
('2024-05-14 19:45:00', 36.99, 5.9, 'alessia.martini@example.com'),
('2024-05-14 20:20:00', 40.75, 9.2, 'marco.moretti@example.com'),
('2024-05-15 18:00:00', 49.99, 6.8, 'giulia.conti@example.com'),
('2024-05-15 21:15:00', 32.50, 4.5, 'roberto.gallo@example.com');
```

```
INSERT INTO Consegna (ordine, orarioStimato, orarioArrivo, mancia, problema, rider)
```

```
VALUES
```

```
(1, '13:00:00', '13:10:00', 5.00, NULL, 1),
(2, '19:30:00', '19:45:00', 0.00, 'consegna non effettuata', 2),
(3, '20:15:00', '20:30:00', 3.50, NULL, 3),
(4, '18:00:00', '18:20:00', 0.00, 'netto ritardo', 4),
(5, '13:45:00', '13:55:00', 2.00, NULL, 5),
(6, '21:00:00', '21:20:00', 0.00, NULL, 6),
(7, '20:00:00', '20:15:00', 4.00, NULL, 7),
(8, '20:30:00', '20:45:00', 0.00, 'reclamo', 8),
(9, '19:15:00', '19:25:00', 3.00, NULL, 9),
(10, '21:45:00', '22:00:00', 0.00, NULL, 10);|
```

```
INSERT INTO OrdinePiatto (piatto, ordine, quantità)
VALUES
(1, 1, 2),
(2, 1, 1),
(3, 2, 1),
(4, 2, 2),
(5, 3, 1),
(6, 4, 1),
(7, 4, 1),
(8, 5, 3),
(9, 5, 1),
(10, 6, 1);|
```

```
INSERT INTO IngredientePiatto (piatto, ingrediente)
VALUES
(1, 'Pomodoro'),
(1, 'Mozzarella'),
(1, 'Basilico'),
(2, 'Parmigiano');|
```

```
INSERT INTO CategoriaPiatto (piatto, categoria)
VALUES
(1, 'Salato'),
(1, 'Più venduto'),
(2, 'Salato'),
(3, 'Salato'),
(4, 'Salato'),
(5, 'Salato'),
(6, 'Salato'),
(10, 'Dolce');|
```

```
INSERT INTO AllergenePiatto (piatto, allergene)
VALUES
(1, 'Glutine'),
(2, 'Lattosio'),
(4, 'Glutine');
```

```
INSERT INTO CodiceSconto (percentuale, soldi, borsellino, ordine)
VALUES
(20, NULL, 'mario.rossi@example.com', 1), -- Sconto percentuale del 20%
(NULL, 5, 'laura.bianchi@example.com', 2), -- Sconto di 5 euro
(15, NULL, 'luigi.verdi@example.com', 3), -- Sconto percentuale del 15%
(NULL, 8, 'giovanna.russo@example.com', 4), -- Sconto di 8 euro
(30, NULL, 'francesca.ferrari@example.com', 5), -- Sconto percentuale del 30%
(NULL, 10, 'paolo.ricci@example.com', 6), -- Sconto di 10 euro
(25, NULL, 'alessia.martini@example.com', 7), -- Sconto percentuale del 25%
(NULL, 6, 'marco.moretti@example.com', 8), -- Sconto di 6 euro
(10, NULL, 'giulia.conti@example.com', 9), -- Sconto percentuale del 10%
(NULL, 12, 'roberto.gallo@example.com', 10); -- Sconto di 12 euro|
```



```
INSERT INTO ChatRistorante (consegna)
VALUES
```

```
(1), -- Associa una chat alla consegna con id 1
(2), -- Associa una chat alla consegna con id 2
(3), -- Associa una chat alla consegna con id 3
(4), -- Associa una chat alla consegna con id 4
(5), -- Associa una chat alla consegna con id 5
(6), -- Associa una chat alla consegna con id 6
(7), -- Associa una chat alla consegna con id 7
(8), -- Associa una chat alla consegna con id 8
(9), -- Associa una chat alla consegna con id 9
(10); -- Associa una chat alla consegna con id 10|
```

```
INSERT INTO ChatRider (consegna)
VALUES
```

```
(1), -- Associa una chat alla consegna con id 1
(2), -- Associa una chat alla consegna con id 2
(3), -- Associa una chat alla consegna con id 3
(8), -- Associa una chat alla consegna con id 8
(10); -- Associa una chat alla consegna con id 10|
```

```
INSERT INTO MessaggioRistorante (testo, DataOra, utente, ristorante, chatRistorante)
VALUES
```

```
('Ordine in lavorazione', CURRENT_TIMESTAMP, 'mario.rossi@example.com', 1, 1),
('Il tuo ordine è in arrivo!', CURRENT_TIMESTAMP, 'laura.bianchi@example.com', 2, 2),
('Consegna prevista per le 19:30', CURRENT_TIMESTAMP, 'luigi.verdi@example.com', 3, 3),
('Gentile cliente, il suo ordine è in ritardo di 10 minuti', CURRENT_TIMESTAMP, 'giovanna.russo@example.com', 4, 4),
('Stiamo preparando il suo ordine', CURRENT_TIMESTAMP, 'francesca.ferrari@example.com', 5, 5),
('Il rider è in viaggio', CURRENT_TIMESTAMP, 'paolo.ricci@example.com', 6, 6),
('Il suo ordine è stato consegnato', CURRENT_TIMESTAMP, 'alessia.martini@example.com', 7, 7),
('Grazie per aver ordinato da noi!', CURRENT_TIMESTAMP, 'marco.moretti@example.com', 8, 8),
('Ordine in lavorazione', CURRENT_TIMESTAMP, 'giulia.conti@example.com', 9, 9),
('Il suo ordine è stato consegnato', CURRENT_TIMESTAMP, 'roberto.gallo@example.com', 10, 10),
('Ciao, vorrei apportare una modifica al mio ordine.', CURRENT_TIMESTAMP, 'mario.rossi@example.com', 1, 1),
('Certamente, quale modifica desidera fare?', CURRENT_TIMESTAMP, 'laura.bianchi@example.com', 2, 2),
('Posso aggiungere un dessert al mio ordine?', CURRENT_TIMESTAMP, 'luigi.verdi@example.com', 3, 3),
('Mi scusi, ma il mio ordine sembra essere in ritardo.', CURRENT_TIMESTAMP, 'giovanna.russo@example.com', 4, 4),
('È possibile avere una stima più precisa dell'orario di consegna?', CURRENT_TIMESTAMP, 'francesca.ferrari@example.com', 5, 5),
('Grazie per la conferma.', CURRENT_TIMESTAMP, 'paolo.ricci@example.com', 6, 6),
('Il mio ordine è arrivato correttamente. Grazie!', CURRENT_TIMESTAMP, 'alessia.martini@example.com', 7, 7),
('È stato un piacere, speriamo di rivederla presto!', CURRENT_TIMESTAMP, 'marco.moretti@example.com', 8, 8),
('Ho appena ricevuto una notifica di ritardo sul mio ordine.', CURRENT_TIMESTAMP, 'giulia.conti@example.com', 9, 9),
('Mi scusi, ma il mio ordine sembra essere incompleto.', CURRENT_TIMESTAMP, 'roberto.gallo@example.com', 10, 10);|
```

```
INSERT INTO MessaggioRider (testo, dataOra, utente, rider, chatRider)
```

```
VALUES
```

```
('Il mio ordine è in ritardo', CURRENT_TIMESTAMP, 'mario.rossi@example.com', 1, 1),
('Sto avendo problemi tecnici con il mio mezzo', CURRENT_TIMESTAMP, 'laura.bianchi@example.com', 2, 2),
('Sono in viaggio, arrivo tra 10 minuti', CURRENT_TIMESTAMP, 'luigi.verdi@example.com', 3, 3),
('Mi dispiace, ci sono dei ritardi nella consegna', CURRENT_TIMESTAMP, 'giovanna.russo@example.com', 4, 4),
('Sto aspettando in zona per la consegna', CURRENT_TIMESTAMP, 'francesca.ferrari@example.com', 5, 5),
('Mi sono perso, sto cercando la strada giusta', CURRENT_TIMESTAMP, 'paolo.ricci@example.com', 6, 1),
('Ho avuto un contrattempo, arrivo il prima possibile', CURRENT_TIMESTAMP, 'alessia.martini@example.com', 7, 2),
('Ho appena ritirato l'ordine, sto arrivando', CURRENT_TIMESTAMP, 'marco.moretti@example.com', 8, 3),
('Mi scusi, sono in ritardo di 5 minuti', CURRENT_TIMESTAMP, 'giulia.conti@example.com', 9, 4),
('Il traffico è molto intenso, sto cercando di accelerare', CURRENT_TIMESTAMP, 'roberto.gallo@example.com', 10, 5),
('L'ordine è stato consegnato con successo', CURRENT_TIMESTAMP, 'mario.rossi@example.com', 1, 1),
('Grazie per la cortesia', CURRENT_TIMESTAMP, 'laura.bianchi@example.com', 2, 2),
('Apprezzo la puntualità nella consegna', CURRENT_TIMESTAMP, 'luigi.verdi@example.com', 3, 3),
('Mi scusi per il ritardo, è stato imprevisto', CURRENT_TIMESTAMP, 'giovanna.russo@example.com', 4, 4),
('Tutto bene, grazie per la consegna', CURRENT_TIMESTAMP, 'francesca.ferrari@example.com', 5, 5),
('Grazie per l'update sulla consegna', CURRENT_TIMESTAMP, 'paolo.ricci@example.com', 6, 1),
('È arrivato il mio ordine, tutto perfetto', CURRENT_TIMESTAMP, 'alessia.martini@example.com', 7, 2),
('Grazie per la comunicazione tempestiva', CURRENT_TIMESTAMP, 'marco.moretti@example.com', 8, 3),
('Non c'è problema, grazie per l'avviso', CURRENT_TIMESTAMP, 'giulia.conti@example.com', 9, 4),
('Capita, grazie per l'aggiornamento', CURRENT_TIMESTAMP, 'roberto.gallo@example.com', 10, 5);|
```



```
INSERT INTO RecensioneRider (testo, stelle, consegna)
VALUES
(NULL, 5, 1),
('Servizio eccellente, il rider ha consegnato l'ordine in anticipo.', 4, 2),
('Il rider era cortese ma l'ordine è arrivato leggermente freddo.', 3, 3),
('Consegna un po' in ritardo, ma il rider si è scusato e il cibo era ancora caldo.', 4, 4),
('Il rider era simpatico ma l'ordine era incompleto.', 2, 5),
('Arrivato presto, gentile e il cibo era caldo!', 5, 6),
('Rider molto cordiale e professionale.', 5, 7),
('Purtroppo il mio ordine è stato consegnato in ritardo e il cibo era freddo.', 2, 8),
('Servizio veloce e cortese.', 4, 9),
('Il rider era gentile ma l'ordine è arrivato in ritardo.', 3, 10);
```

```
INSERT INTO RecensioneRistorante (testo, stelle, consegna)
VALUES
('Ottimo cibo e servizio impeccabile.', 5, 1),
('Il ristorante ha superato le mie aspettative.', 4, 2),
('Cibo delizioso e consegna veloce.', 5, 3),
('Il cibo era mediocre.', 3, 4),
('Ristorante consigliato, cibo fresco e gustoso.', 4, 5),
('Il ristorante ha deluso le aspettative, cibo freddo e consegna in ritardo.', 2, 6),
('Servizio eccellente e piatti gustosi.', 5, 7),
('Ristorante molto pulito e cibo di alta qualità.', 4, 8),
('Cibo delizioso, un po' freddo', 3, 9),
('Esperienza complessivamente positiva, buon cibo', 5, 10);
```

3.3 Qualche operazione di cancellazione e modifica

3.3.1 Operazioni di cancellazione

- Cancellazione di un di un determinato Piatto:

```
DELETE FROM Piatto  
WHERE titolo LIKE 'Pizza Margherita'
```

Cancellando la tupla nella tabella Piatto, che ha come valore dell'attributo titolo 'Pizza Margherita', non vengono cancellate altre tuple nel database perchè nessun attributo ha un vincolo d'integrità referenziale con la tabella Piatto.

3.3.2 Operazioni di modifica

- Modifica dell'attributo email nella tabella Utente:

```
UPDATE Utente  
SET email = 'mariorossi84@gmail.com'  
WHERE email LIKE 'mario.rossi@example.com'
```

Aggiornando l'attributo email nella tabella Utente, vengono aggiornate di conseguenza tutte le email delle tabelle esterne che hanno un vincolo d'integrità referenziale con l'attributo email in Utente per via della clausola on update cascade.