



Progetto di Piattaforma di Food Delivery  
Laboratorio Basi di Dati 2023/2024

Agnese Andriani

May 30, 2024

# Contents

<b>1</b>	<b>Progettazione Concettuale</b>	<b>2</b>
1.1	Requisiti Iniziali . . . . .	2
1.2	Glossario dei Termini . . . . .	6
1.3	Requisiti rivisti e strutturati in gruppi di frasi omogenee . . . . .	7
1.3.1	Requisiti rivisti . . . . .	7
1.3.2	Requisiti strutturati in gruppi di frasi omogenee . . . . .	8
1.4	Schema E-R + Business Rules . . . . .	11
<b>2</b>	<b>Progettazione Logica</b>	<b>12</b>
2.1	Tavola dei Volumi . . . . .	12
2.2	Tavola delle Operazioni . . . . .	13
2.3	Ristrutturazione dello schema E-R . . . . .	15
2.3.1	Analisi delle ridondanze . . . . .	15
2.3.2	Eliminazione delle generalizzazioni . . . . .	17
2.3.3	Eliminazione degli attributi composti . . . . .	20
2.4	Schema E-R ristrutturato + Business Rules . . . . .	21
2.4.1	Business Rules . . . . .	22
2.5	Schema Relazionale . . . . .	24
<b>3</b>	<b>Implementazione</b>	<b>25</b>
3.1	Creazione dei Database . . . . .	25
3.2	Popolamento di Tutte le Tabelle del Database . . . . .	28

# Chapter 1

## Progettazione Concettuale

### 1.1 Requisiti Iniziali

Si vuole progettare la base di dati per un 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.

Gli utenti si registrano al servizio inserendo nome, email, password, numero di telefono, indirizzo di recapito. Inoltre, l'utente deve indicare un metodo di pagamento e ricaricare il proprio borsellino elettronico che possiede un proprio saldo. L'utente può possedere codici sconto.

Ogni ristorante (Figura 1) è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un'immagine di profilo e un numero di stellette. Ogni ristorante appartiene a una o più categorie in base al tipo di cibo offerto. I ristoranti che possono diventare Top partner se soddisfano i seguenti requisiti:

- Devono avere almeno 20 ordini consegnati correttamente
- Devono avere una valutazione clienti maggiore o uguale a 4.5 stelline su cinque
- Devono avere una percentuale massima di ordini annullati dal ristorante dell'1.5
- Devono avere una percentuale massima di ordini con reclami del 2.5

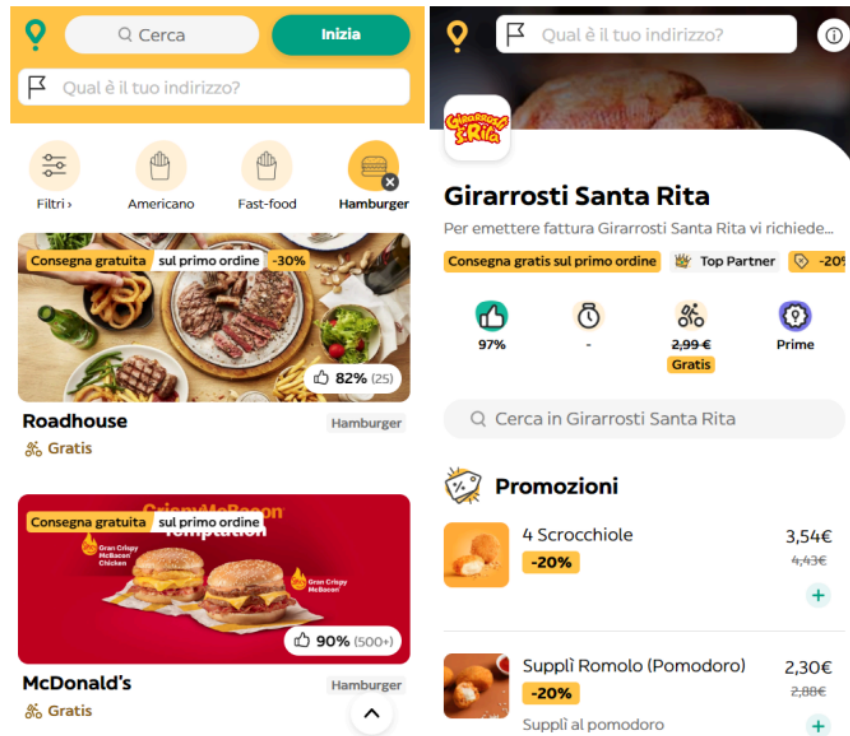
Inoltre, si tiene traccia della data in cui questi ristoranti sono entrati a far parte della categoria.

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.

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.

Gli utenti possono eventualmente effettuare dei reclami inviando un messaggio al ristorante.



Si gestisce un numero arbitrario di riders. 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 (bici normale, bici elettrica, monopattino). I riders che utilizzano il monopattino devono indicare quanti km possono effettuare.

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 superiore ai 10 km, solo i rider con bici elettrica vengono scelti. Si tiene 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.

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. La base di dati deve supportare una volta al mese l'aggiornamento delle 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	Una persona che utilizza il servizio Cibora per ordinare cibo. Deve registrarsi con nome, email, password, telefono e indirizzo. Deve scegliere un metodo di pagamento.	Cliente	Ristorante Ordine Recensione Piatto Chat Borsellino
Borsellino	Necessario ad un utente per poter ordinare. Possiede un saldo. Il borsellino viene ricaricato dall'utente stesso.		Utente
Ristorante	Un'attività che offre cibo attraverso il servizio Cibora. Fornisce informazioni come nome, descrizione, indirizzo, costo di spedizione, immagine di profilo, numero di stellette.		Utente Piatto Categoria
Piatto	Un prodotto offerto da un ristorante. Ha titolo, immagine, ingredienti, allergeni, prezzo e eventuale sconto.	Portata Cibo Pietanze	Ristorante Ordine
Ordine	Una richiesta da parte di un utente per un insieme di piatti da un ristorante. Include informazioni come data, stato, rider assegnato.		Utente Ristorante Piatto Rider
Chat	Un sistema di comunicazione tra utenti e ristoranti o utenti e rider.		Utente
Recensione	Valutazione lasciata dagli utenti dopo una consegna. Può includere una valutazione da 1 a 5 stelle e un commento testuale opzionale.	Valutazione	Utente
Rider	Persona che consegna gli ordini. È identificato da un codice e ha uno stato (occupato, disponibile, fuori servizio).	Fattorino	Stato Ordine Mezzo
Mezzo	Tipo di veicolo utilizzato dai rider per consegnare gli ordini. Può essere bicicletta normale, bicicletta elettrica, o monopattino.		Rider
Stato	Indicatore della disponibilità del rider. Può essere "occupato", "disponibile", o "fuori servizio".	Disponibilità	Rider
Categoria	Gruppo a cui può appartenere un ristorante, in base al tipo di cibo offerto. Esempi di categorie includono "Fast Food", "Vegetariano", ecc.		Ristorante
Allergeni	Sostanza contenuta nei piatti offerti a cui gli utenti potrebbero essere allergici. Ogni piatto contiene la lista di questi allergeni.		Piatto
Ingredienti	Componenti con cui si prepara un piatto.		Piatto
Lista	Gruppo a cui può appartenere un piatto. Esempi di liste includono i più venduti, promozioni, dolci, salato, ecc.)		Piatto

## 1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

### 1.3.1 Requisiti rivisti

Si vuole realizzare una base di dati per un servizio di food delivery che permette di gestire i dati degli utenti, dei ristoranti e dei riders.

Per il dato utente registriamo: nome, email, password, nome, numero di telefono, indirizzo di recapito, metodo di pagamento, saldo del borsellino e eventuali sconti.

Per il dato borsellino registriamo: saldo e utente a cui appartiene il borsellino elettronico.

Per ogni utente registriamo: se è un utente premium o meno.

Per il dato ordine registriamo: data, ora e lo stato. Ogni utente può avere uno o più ordini. L'ordine viene ricevuto dal ristorante interessato.

Per il dato ristorante registriamo: nome, indirizzo, costo spedizione, immagine di profilo, una descrizione e il numero di stelle. Per ogni ristorante sono scritte zero o più recensioni. Ogni ristorante offre uno o più piatti. Ogni ristorante appartiene a zero o più categorie.

Per il dato categoria registriamo: nome.

I ristoranti per diventare top partner devono soddisfare i seguenti requisiti:

- Devono avere almeno 20 ordini consegnati correttamente
- Devono avere una valutazione clienti maggiore o uguale a 4.5 stelline su cinque
- Devono avere una percentuale massima di ordini annullati dal ristorante dell'1.5
- Devono avere una percentuale massima di ordini con reclami del 2.5

Per ogni top partner conosciamo la data di ingresso nella categoria.

Per il dato piatto registriamo: titolo, immagine, prezzo, un eventuale sconto.

Ogni piatto contiene una lista di ingredienti e una lista di allergeni. Per il dato allergeni registriamo: nome. Per il dato ingredienti registriamo: nome.

Ogni ordine è consegnato da un solo rider.



Per il dato rider registriamo: codice, numero di consegne effettuate, posizione GPS, un'eventuale mancia. Ogni rider si sposta con un mezzo. Il mezzo può essere: bicicletta normale, bicicletta elettrica o monopattino. Il rider con monopattino deve riferire i km.

Ogni rider ha uno stato. Lo stato può essere: occupato, disponibile o fuori servizio.

Un utente può decidere se avviare una chat o meno.

Per il dato chat registriamo: data inizio, ora inizio, messaggio. La chat potrà essere utente-ridere e utente-ristorante.

Effettuata la consegna, l'utente può valutare il servizio con una recensione. Per il dato recensione registriamo: data, ora, utente, valutazione e un eventuale commento testuale. La recensione può essere effettuata nei confronti del ristorante e del rider.

La base di dati deve supportare una volta al mese l'aggiornamento delle seguenti classifiche:

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

### **1.3.2 Requisiti strutturati in gruppi di frasi omogenee**

**Frasi relative a Utente:**

- Gli utenti forniscono nome, email, password, numero di telefono e indirizzo di recapito.
- Gli utenti devono associare un mezzo di pagamento.
- Gli utenti possono sottoscrivere la modalità premium, che conferisce priorità sugli ordini.
- Gli utenti possono raccogliere codici di sconto basati sul numero di ordini precedenti.
- Gli utenti possono recensire ristoranti e riders con una valutazione da 1 a 5 e un commento testuale opzionale.
- Gli utenti possono avviare chat con i rider e con i ristoranti in caso di problemi con l'ordine.
- Gli utenti creano liste di piatti da ordinare.

- Gli utenti possono inviare reclami ai ristoranti per ordini passati.

**Frase relative a Ristorante:**

- Ogni un borsellino elettronico ha un saldo.
- Il borsellino può essere ricaricato in qualsiasi momento dall'utente.

**Frase relative a Ristorante:**

- Ogni ristorante deve fornire nome, descrizione, indirizzo, costo di spedizione, immagine di profilo, e numero di stelle.
- Il numero di stelle è aggiornato ogni lunedì, basato sulla percentuale di recensioni positive dell'ultima settimana.
- Ogni ristorante deve appartenere a una o più categorie in base al tipo di cibo offerto.
- I ristoranti propongono agli utenti una lista di piatti da ordinare.
- Un ristorante è considerato "Top Partner" se ha:
  - Almeno 20 ordini consegnati correttamente.
  - Valutazione clienti di almeno 4.5 stelle su 5.
  - Meno dell'1.5
  - Meno del 2.5
- I Top Partner compaiono in una sezione dedicata dell'app e ricevono un badge speciale.
- Per i Top Partner, si tiene traccia della data di inclusione nella categoria.

**Frase relative a Piatto:**

- Ogni piatto deve avere titolo, immagine, prezzo, e un eventuale sconto.

**Frase relative a Lista:**

- I piatti possono appartenere a più liste (es. più venduti, promozioni, dolci, salato).

**Frase relative a Ingredienti:**

- Ogni portata ha una lista di ingredienti.

**Frase relative a Allergeni:**

- Ogni portata ha una lista di allergeni.

**Frase relative a Ordine:**

- Gli ordini possono essere annullati da utenti o ristoratori, a meno che non siano stati affidati a un rider per la consegna.

**Frase relative a Rider:**

- Ogni rider è identificato da un codice e ha uno stato che può essere "disponibile", "occupato", o "fuori servizio", dalla posizione aggiornata in tempo reale attraverso GPS.
- Ogni rider è classificato secondo il tipo di mezzo (bicicletta normale, bicicletta elettrica, monopattino).
- I rider con monopattino devono indicare quanti km possono effettuare prima che la batteria si scarichi.
- I rider con biciclette elettriche sono coinvolti per ordini con tragitto superiore a 10 km.
- Ogni mese vengono pubblicate le classifiche dei rider più veloci nel consegnare gli ordini.
- Si tiene traccia del numero di consegne effettuate da ogni rider e dei tempi di consegna.

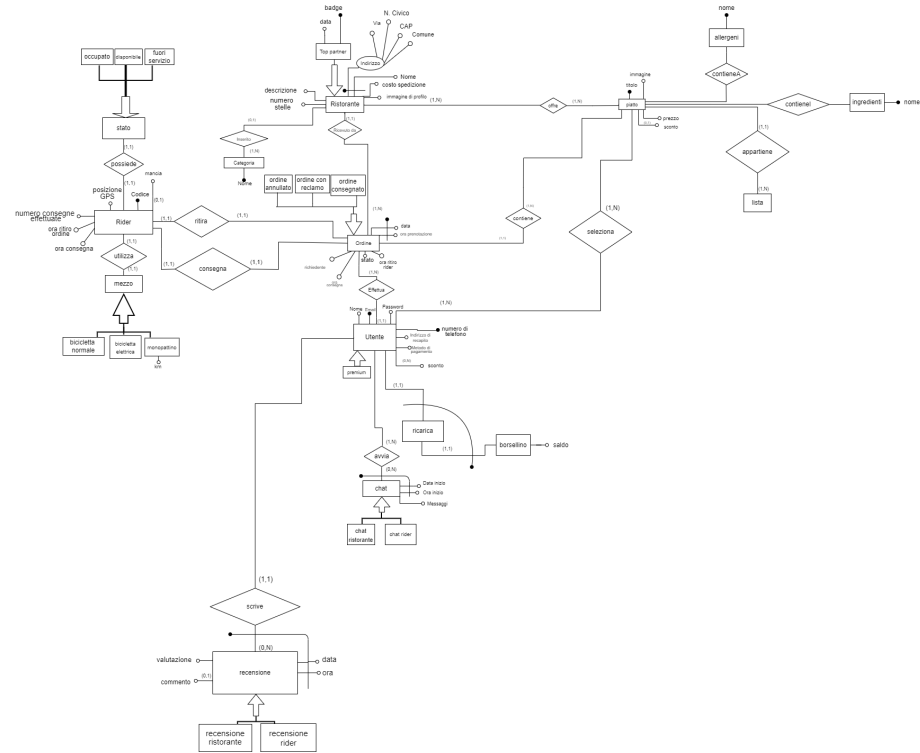
**Frase relative a Recensione:**

- Le recensioni sono una valutazione da 1 a 5 stelle.
- Una recensione può contenere un commento testuale.

**Frase relative a Chat:**

- Possono essere avviate dagli utenti con i ristoranti e con i rider per discutere eventuali problemi relativi agli ordini.
- Le chat possono essere utilizzate dagli utenti per inviare reclami o segnalare problemi con l'ordine.

## 1.4 Schema E-R + Business Rules



## Chapter 2

# Progettazione Logica

### 2.1 Tavola dei Volumi

Concetto	Tipo	Volume	Motivazione
Utente	E	10.000	Ipotizziamo una piattaforma in cui sono iscritti 10.000 di utenti.
Ristoranti	E	200	Ipotizziamo che sulla piattaforma si iscrivono circa 200 ristoranti.
Piatti	E	1.000	Ipotizziamo che nella piattaforma verranno registrati circa mille piatti. Ogni ristorante offre una media di 5 piatti.
Ordini	E	20.000	Ipotizziamo che sulla piattaforma siano stati effettuati circa 2 ordini al giorno per utente.
Rider	E	150	Ipotizziamo che ci siano circa 150 rider che lavorano per la piattaforma.
Recensione	E	5.000	Ipotizziamo che sulla piattaforma vengano scritte circa 5.000 recensioni. Vengono incluse sia quelle ai ristoranti sia quelle ai rider.
Chat	E	15.000	Ipotizziamo che sulla piattaforma vengano scritti circa 15.000 commenti
Lista	E	50	Ipotizziamo che sulla piattaforma vengano create circa 50 liste di piatti
Mezzo	E	150	Ipotizziamo che ci siano in totale 150 mezzi utilizzati dai rider.
Allergeni	E	14	Ipotizziamo che ci siano in totale 14 allergeni nei piatti.
Ingredienti	E	50	Ipotizziamo che si possano usare in totale 50 ingredienti.
Continua all pagina successiva			

Concetto	Tipo	Volume	Motivazione
Borsellino	E	10.000	Ipotizziamo che ogni utente abbia un borsellino.
Categoria	E	100	Ipotizziamo che vengano offerte 100 categorie di ristoranti.
Scrivo	R	5.000	Ipotizziamo che vengano scritte 5.000 recensioni.
Avvia	R	8.000	Gli utenti possono avviare chat con ristoranti o rider.
Effettua	R	20.000	Numero totale di ordini effettuati dagli utenti sulla piattaforma.
Appartiene	R	1.000	Ogni piatto appartiene a una lista.
Contiene	R	40.000	Ogni ordine può contenere più piatti
Ricevuto	R	20.000	Gli ordini sono ricevuti dai ristoranti.
Offre	R	1.000	I ristoranti offrono piatti agli utenti.
Utilizza	R	150	Ogni rider utilizza un mezzo per spostarsi.
Possiede	R	150	Ogni rider possiede uno stato.
Ritira	R	2.000	Ogni rider ritira dai ristoranti gli ordini.
Consegna	R	2.000	Ogni rider può consegnare agli utenti.
Ricarica	R	10.000	Ogni utente ricarica il borsellino.
Seleziona	R	500	Vengono selezionati in totale 500 piatti.
ContieneI	R	50	Gli ingredienti contenuti nei piatti, in totale, sono 50.
ContieneA	R	14	Gli allergeni contenuti nei piatti, in totale, sono 14.
Inserito	R	100	Ogni ristorante viene inserito in una o più categorie.

## 2.2 Tavola delle Operazioni

#	Operazione	Tipo	Frequenza	Motivazione
1	Registrazione di nuovi utenti	I	200/day	Operazione iniziale della piattaforma che permette ad una persona di utilizzare il servizio.
2	Aggiornamento del profilo utente	A	100/day	Operazione che permette il cambiamento di dati esistenti come nome, email, numero di telefono, ecc.
Continua alla pagina successiva				

#	Operazione	Tipo	Frequenza	Motivazione
3	Creazione di nuovi ordini	I	2.000/day	Operazione che permette all'utente di effettuare un ordine con uno o più piatti.
4	Aggiornamento dello stato dell'ordine	A	3.000/day	Operazione che permette di aggiornare lo stato dell'ordine come "in lavorazione", "in consegna", "consegnato".
5	Aggiornamento del borsellino	A	200/day	Operazione che avviene quando un utente ricarica il saldo del proprio borsellino elettronico.
6	Scrittura di un commento	I	13.000/day	Operazione che permette ad un host e/o un utente di lasciare ulteriori feedback testuali ad una recensione
7	Chat tra utenti e rider	L/S	1.500/day	Operazione che permette la comunicazione tra utenti, ristoranti e rider tramite chat.
8	Assegnazione di rider agli ordini	A	2.000/day	Operazione che permette di assegnare un ordine ad un rider.
9	Consegna degli ordini	A	2.000/week	Operazione che avviene quando un rider consegna un ordine.
10	Lasciare recensioni	I	500/day	Operazione che permette agli utenti di scrivere una recensione di un ristorante o di un rider.
11	Annullamento degli ordini	C	200/day	Operazione che permette ad un utente o ad un ristorante di annullare un ordine.
12	Aggiunta di piatti ai ristoranti	I	100/day	Operazione che permette ai ristoranti di aggiungere nuovi piatti a quelli già presenti.
13	Raccolta di codici sconto	I	200/day	Operazione che permette agli utenti di raccogliere codici sconto in base agli ordini effettuati in passato.
14	Ritiro ordini	A	2.000/day	Operazione che permette ai rider di ritirare gli ordini dai ristoranti per poterli consegnare.

## 2.3 Ristrutturazione dello schema E-R

### 2.3.1 Analisi delle ridondanze

#### Attributo "Ora ritiro rider" e "Ora consegna"

All'interno dello schema ER, gli attributi "Ora ritiro rider" e "Ora consegna" in Rider vengono considerati ridondanti in quanto esistono già gli attributi corrispondenti "Ora Ritiro Ordine" e "Ora Consegna" in Ordine. Questi attributi indicano rispettivamente l'orario in cui il rider ritira l'ordine presso il ristorante e l'orario in cui consegna l'ordine al cliente.

#	Concetto	Tipo	Volume	Motivazione
1	Ordine	A	2.000/day	Operazione principale che permette di effettuare un soggiorno

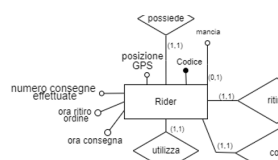


Figure 2.1: Con ridondanza

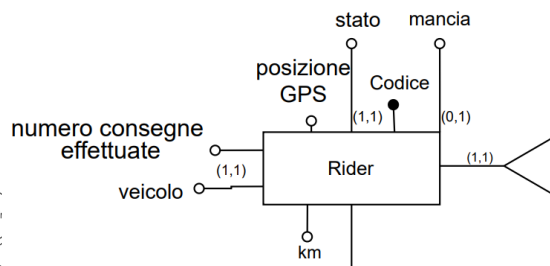


Figure 2.2: Senza ridondanza



### Tavola degli accessi

Analizziamo l'operazione "Creazione di nuovi ordini (2.000 volte al giorno)"

In presenza di ridondanza

Concetto	Costrutto	Accessi	Tipo
Ordine	E	1	L
Ordine	E	1	S

Senza la ridondanza

Concetto	Costrutto	Accessi	Tipo
Ordine	E	1	S
Ordine	E	1	L
Ritira	R	1	L
Consegna	R	1	L

Analisi di complessità in presenza di ridondanza:

- In termini di tempo
  - vengono effettuati un accesso in lettura ed uno in scrittura, per un totale di 4.000 accessi (contiamo doppi gli accessi in scrittura)
- In termini di spazio
  - Gli attributi ridondati "ora ritiro rider" e "ora consegna rider" occupano 8 byte ciascuno, per un totale di 32 Kb (32.000 byte).

Analisi di complessità in assenza di ridondanza:

- In termini di spazio
  - il costo totale è 0 byte.
- In termini di tempo
  - vengono effettuati due accessi in lettura e due in scrittura, per un totale di 8.000 accessi.

Dall'analisi effettuata, con l'assenza di ridondanza, risulta un peggioramento nei tempi di accesso tuttavia, essendo un'operazione che risulta essere essenziale in un servizio di food delivery, si potrebbe preferire mantenere la ridondanza per garantire tempi di risposta più veloci agli utenti. preferiamo mantenere la ridondanza.

### 2.3.2 Eliminazione delle generalizzazioni

#### Entità STATO

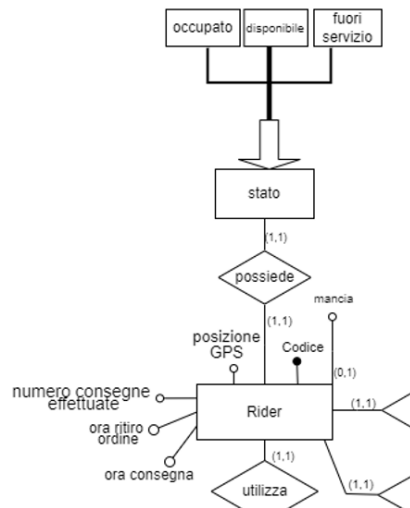


Figure 2.3: Prima

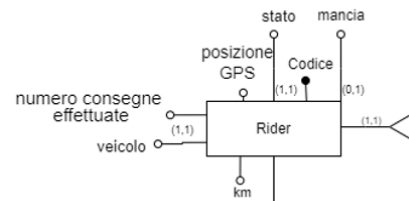


Figure 2.4: Dopo

La generalizzazione è di tipo **totale ed esclusiva**. La decisione consiste nel raggruppamento delle entità figlie nell'attributo stato. L'attributo stato sarà un valore not null.

## Entità MEZZO

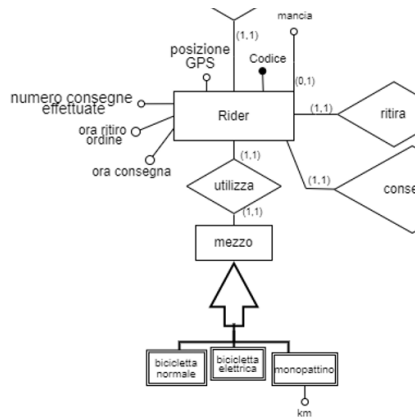


Figure 2.5: Prima

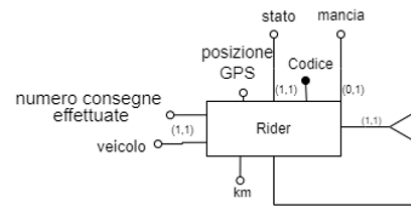


Figure 2.6: Dopo

La generalizzazione è di tipo **totale ed esclusiva**. La decisione consiste nel raggruppamento delle entità figlie nell'attributo veicolo, con valore not null. Gli attributi specifici delle entità figlie (km) vengono spostati nell'entità padre, diventando annullabili

## Entità RECENSIONE

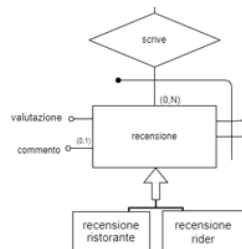


Figure 2.7: Prima

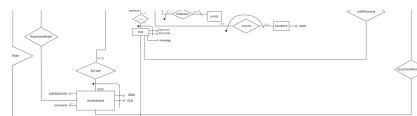


Figure 2.8: Dopo

La generalizzazione è di tipo **parziale e sovrapposta**. La decisione consiste nel sostituire la generalizzazione con associazioni che coinvolgono Rider e Ristorante.

## Entità CHAT

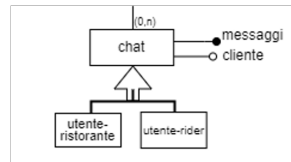


Figure 2.9: Prima

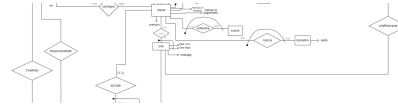


Figure 2.10: Dopo

La generalizzazione è di tipo **parziale e sovrapposta**. La decisione consiste nel sostituire la generalizzazione con associazioni che coinvolgono Rider e Ristorante.

## Entità UTENTE

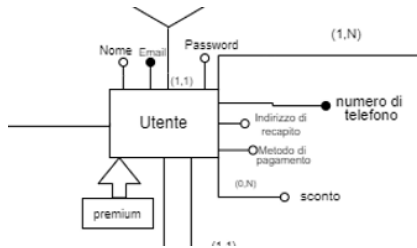


Figure 2.11: Prima

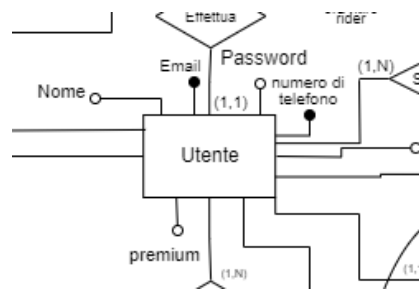


Figure 2.12: Dopo

La generalizzazione è di tipo **parziale e sovrapposta**. La decisione consiste nel raggruppamento dell'entità figlia nell'attributo premium, Il valore può essere null o not null.

## Entità RISTORANTE

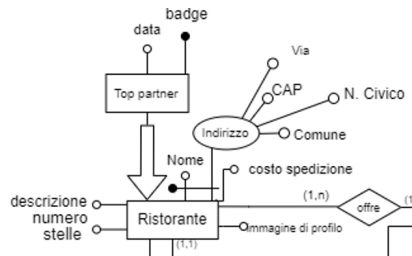


Figure 2.13: Prima

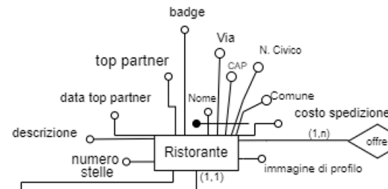


Figure 2.14: Dopo

La generalizzazione è di tipo **parziale e sovrapposta**. La decisione consiste nel raggruppamento dell'entità figlia nell'attributo top partner, con valore not null. Gli attributi specifici delle entità figlie (badge, data top partner) vengono spostati nell'entità padre, diventando annullabili.

## Partizionamento Entità UTENTE

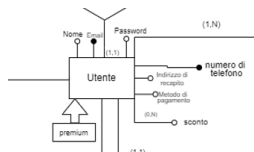


Figure 2.15: Prima

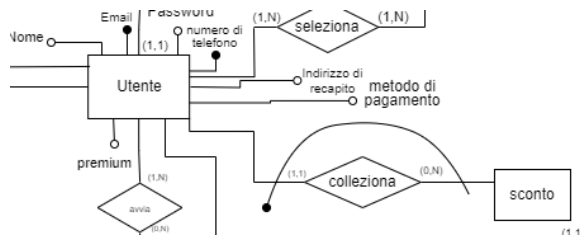


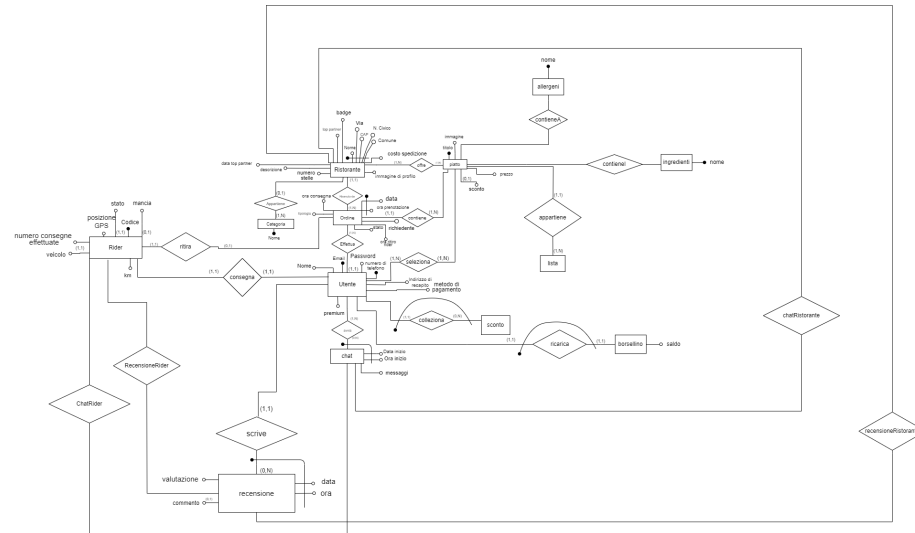
Figure 2.16: Dopo

Decidiamo di partizionare l'entità Utente estraendo l'attributo sconto, facendolo diventare una nuova entità Sconto, associata a Utente tramite la relazione Colleziona.

### 2.3.3 Eliminazione degli attributi composti

L'attributo composto **"indirizzo"** viene eliminato, considerando i suoi componenti come attributi semplici. Nel caso di indirizzo abbiamo: via, numero civico, cap e comune.

## 2.4 Schema E-R ristrutturato + Business Rules



### 2.4.1 Business Rules

#### Derivate dal testo

- Per poter beneficiare del servizio, ogni utente deve registrarsi fornendo nome, email, password, numero di telefono e indirizzo di recapito.
- Ogni utente deve inserire un mezzo di pagamento e ricaricare il borsellino elettronico per poter effettuare un ordine.
- Un ristorante può diventare top partner se dimostra di aver consegnato correttamente almeno 20 ordini, se ha una valutazione clienti maggiore o uguale a 4.5 stelline su cinque, se ha una percentuale massima di ordini annullati dal ristorante dell'1.5 e se ha una percentuale massima di ordini con reclami del 2.5.
- Un utente può visualizzare l'indirizzo di un alloggio solamente dopo aver effettuato una prenotazione con avvenuta conferma da parte dell'host
- Un utente non può indicare in una prenotazione nomi di altri ospiti, se questi non sono iscritti alla piattaforma
- Un utente può valutare un ristorante o un rider solamente quando l'ordine è stato consegnato.
- Un utente non può cancellare un ordine se quest'ultimo è stato affidato ad un rider.
- Un ristorante non può cancellare un ordine se quest'ultimo è stato affidato ad un rider.
- Per distanze superiori ai 10 km vengono interpellati solo i rider con bici elettrica.

#### Introdotte

- L'indirizzo e-mail di un utente che effettua la registrazione deve corrispondere ad un formato e-mail valido.
- Un utente può cancellare solamente i propri ordini.
- Un ristorante può confermare o rifiutare solamente gli ordini ricevuti da loro stessi.
- Un utente non può inserire, in un ordine, un piatto che è compreso tra i piatti offerti da un altro ristorante.
- Un utente non può valutare ordini non effettuati da lui.
- Un utente non può avviare una chat con rider che non consegnano il proprio ordine.

- Un utente non può scrivere commenti sotto recensioni altrui
- Una ordine non può essere cancellato se la data di inizio è antecedente alla data di oggi.



## 2.5 Schema Relazionale

- Utente(Email, Numero telefono, Nome, Password, Indirizzo di recapito, metodo di pagamento, saldo borsellino, premium)
- Sconto(Utente)
  - Sconto(Utente) **referenzia** Utente(Email, Numero telefono)
- Chat(Cliente, Destinatario, Messaggi)
  - Chat(Cliente) **referenzia** Utente(Email, Numero telefono)
- Ordine(ID Ordine, Data, Ora, Stato, Richiedente)
  - Ordine(Richiedente) **referenzia** Utente(Email, Numero telefono)
- Ristorante(ID ristorante, Nome, Via, CAP, Numero Civico, Comune, Costo Spedizione, Immagine di profilo, Numero stelle, Descrizione, Top Partner, Data Top Partner, Badge, ordini consegnati, ordini con reclami, ordini annullati)
- Recensione(ID Recensione, Autore, Data, Ora, Destinatario, Valutazione)
  - Recensione(Autore) **referenzia** Utente(Email, Numero telefono)
- Commento(ID Commento, Data, ID Recensione, Autore, Corpo)
  - Commento(Autore) **referenzia** Utente(Email, Numero telefono)
  - Commento(ID Recensione) **referenzia** Recensione(ID Recensione)
- Categoria(Nome, ID Ristorante)
  - Categoria(ID Ristorante) **referenzia** Ristorante(ID Ristorante)
- Rider(Codice, Mancias, Stato, Posizione GPS, Numero consegne effettuate, Veicolo, Km)
- Piatto(Titolo, Immagine, Lista ingredienti, Lista allergeni, Prezzo, Sconto, ID Ristorante)
  - Piatto(ID Ristorante) **referenzia** Ristorante(ID Ristorante)

# Chapter 3

## Implementazione

Riportiamo in seguito alcune query significative per il nostro database

### 3.1 Creazione dei Database

```
1 DROP TABLE IF EXISTS Commento CASCADE;
2 DROP TABLE IF EXISTS Recensione CASCADE;
3 DROP TABLE IF EXISTS Chat CASCADE;
4 DROP TABLE IF EXISTS Sconto CASCADE;
5 DROP TABLE IF EXISTS Ordine CASCADE;
6 DROP TABLE IF EXISTS Piatto CASCADE;
7 DROP TABLE IF EXISTS Lista CASCADE;
8 DROP TABLE IF EXISTS Ristorante CASCADE;
9 DROP TABLE IF EXISTS Rider CASCADE;
10 DROP TABLE IF EXISTS Categoria CASCADE;
11 DROP TABLE IF EXISTS Utente CASCADE;

1 -- Creazione tabella Utente
2 CREATE TABLE Utente (
3     Nome VARCHAR(50) NOT NULL,
4     Email VARCHAR(255),
5     NumeroTelefono VARCHAR(20) NOT NULL,
6     Password VARCHAR(255) NOT NULL,
7     IndirizzoRecapito VARCHAR(255) NOT NULL,
8     MetodoPagamento VARCHAR(50) NOT NULL,
9     SaldoBorsellino DECIMAL(10,2) NOT NULL DEFAULT 0, -- Saldo iniziale di
        default 0
10    Premium BOOLEAN NOT NULL DEFAULT FALSE, -- Premium disattivato per default
11    PRIMARY KEY (Email, NumeroTelefono),
12
13    CONSTRAINT user_valid_email CHECK (
14        Email ~* '^[A-Za-z0-9._+%~]+@[A-Za-z0-9.-]+\.[A-Za-z]+$'
15    )
16 );
```

```

1 CREATE TABLE Sconto(
2     Utente VARCHAR(40) PRIMARY KEY,
3     FOREIGN KEY (Utente) REFERENCES Utente(Email)
4     ON DELETE CASCADE
5     ON UPDATE CASCADE
6
7 );

1 CREATE TABLE Ristorante(
2     IDRistorante VARCHAR(10) PRIMARY KEY,
3     Nome VARCHAR(100) NOT NULL,
4     Via VARCHAR(100) NOT NULL,
5     CAP VARCHAR(10) NOT NULL,
6     NumeroCivico VARCHAR(10) NOT NULL,
7     Comune VARCHAR(100) NOT NULL,
8     CostoSpedizione DECIMAL(5, 2) NOT NULL CHECK (CostoSpedizione >= 0),
9     ImmagineProfilo TEXT,
10    NumeroStelle DECIMAL(2, 1) CHECK (NumeroStelle BETWEEN 0 AND 5),
11    Descrizione TEXT,
12    TopPartner BOOLEAN DEFAULT FALSE,
13    DataTopPartner DATE,
14    Badge TEXT,
15    OrdiniConsegnati INT NOT NULL DEFAULT 0,
16    OrdiniConReclami INT NOT NULL DEFAULT 0,
17    OrdiniAnnullati INT NOT NULL DEFAULT 0
18 );

1 CREATE TABLE Rider(
2     Codice VARCHAR(10) PRIMARY KEY,
3     Mancias DECIMAL(10, 2),
4     Stato VARCHAR(20) NOT NULL CHECK (Stato IN
5         ('occupato','disponibile','fuori servizio')),
6     PosizioneGPS VARCHAR(50),
7     NumeroConsegneEffettuate INT,
8     Veicolo VARCHAR(20) NOT NULL CHECK (Veicolo IN ('bicicletta normale',
9         'bicicletta elettrica', 'monopattino')),
10    Km DECIMAL(10, 2)
11 );

1 CREATE TABLE Recensione (
2     IDRecensione VARCHAR(10) PRIMARY KEY,
3     Autore VARCHAR(255) NOT NULL,
4     Data DATE NOT NULL,
5     Ora TIME NOT NULL,
6     Destinatarario VARCHAR(255) NOT NULL CHECK (Destinatario IN ('Ristorante',
7         'Rider')),
8     Valutazione INT NOT NULL CHECK (Valutazione BETWEEN 1 AND 5), --
9         Valutazione tra 1 e 5
10    FOREIGN KEY (Autore) REFERENCES Utente(Email) ON DELETE CASCADE
11 );

1 CREATE TABLE Piatto (
2     Titolo VARCHAR(50) PRIMARY KEY,
3     IDRistorante VARCHAR(10),
4     ListaIngredienti VARCHAR(50) NOT NULL,
5     ListaAllergeni VARCHAR(50) NOT NULL,
6     Prezzo DECIMAL(10,2) NOT NULL,
7     Immagine BYTEA NOT NULL,
8     Sconto BOOLEAN NOT NULL DEFAULT FALSE
9     FOREIGN KEY (IDRistorante) REFERENCES Ristorante(IDRistorante)
10 );

```

```

1 CREATE TABLE Ordine(
2     IDOrdine SERIAL PRIMARY KEY,
3     Data DATE NOT NULL,
4     Ora TIME NOT NULL,
5     StatoOrdine VARCHAR(20) NOT NULL CHECK (Stato IN ('In attesa', 'In
        preparazione', 'Consegnato', 'Annullato')),
6     Richiedente VARCHAR(40) NOT NULL,
7     IDRistorante VARCHAR(10) NOT NULL,
8     FOREIGN KEY (IDRistorante) REFERENCES Ristorante(IDRistorante)
9     FOREIGN KEY (Richiedente) REFERENCES Utente(Email)
10    ON DELETE SET NULL
11    ON UPDATE CASCADE
12 );

```

```

1 CREATE TABLE Commento (
2     IDCommento VARCHAR(10) PRIMARY KEY,
3     Data DATE NOT NULL,
4     IDRecensione INT NOT NULL,
5     Autore VARCHAR(255) NOT NULL,
6     Corpo TEXT,
7     FOREIGN KEY (Autore) REFERENCES Utente(Email) ON DELETE CASCADE,
8     FOREIGN KEY (IDRecensione) REFERENCES Recensione(IDRecensione) ON DELETE
        CASCADE
9 );

```

```

1 CREATE TABLE Chat(
2     Messaggi TEXT,
3     Destinatario VARCHAR(40) NOT NULL CHECK (Destinatario IN ('Ristorante',
        'Rider')),
4     Cliente VARCHAR(40) PRIMARY KEY,
5     FOREIGN KEY (Cliente) REFERENCES Utente(Email)
6     ON DELETE CASCADE
7     ON UPDATE CASCADE
8 );

```

```

1 CREATE TABLE Categoria (
2     Nome VARCHAR(50) PRIMARY KEY
3     IDRistorante VARCHAR(10),
4     FOREIGN KEY (IDRistorante) REFERENCES Ristorante(IDRistorante)
5     ON DELETE CASCADE
6     ON UPDATE CASCADE
7 );

```

## 3.2 Popolamento di Tutte le Tabelle del Database

```
1 INSERT INTO Utente (Nome, Cognome, Email, Password, NumeroTelefono,
2   IndirizzoRecapito, MetodoPagamento, SaldoBorsellino, Premium)
3   VALUES ('Agnese', 'Andriani', 'agnese@email.com', 'password123',
4     '1234567890', 'Via Pessinetto 1', 'Carta di Credito', 100.00, TRUE),
5     ('Mario', 'Rossi', 'mario@email.com', 'password456', '0987654321',
6     'Via Roma 2', 'PayPal', 50.00, FALSE);
7
8 INSERT INTO
9   Ristorante(IDRistorante, Nome, Via, CAP, NumeroCivico, Comune, CostoSpedizione,
10  ImmagineProfilo, NumeroStelle, Descrizione, TopPartner, DataTopPartner, Badge,
11  OrdiniConsegnati,
12  OrdiniConReclami, OrdiniAnnullati)
13  VALUES ('789d', 'Ristorante da Lucia', 'Via Garibaldi',
14    '10140', '15', 'Torino', 2.00,
15    'https://it.wiktionary.org/wiki/pizzeria#/media/File:PIZZERIA_1830.jpg',
16    3,
17    'Ristorante di cucina tradizionale piemontese', FALSE, NULL, NULL, 4,
18    1, 0),
19    ('456U', 'Pizzeria da Ciro', 'Via Napoli', '10139', '19', 'Torino', 3.50,
20    'https://encrypted-tbn0.gstatic.com/images',
21    1.5, 'Pizzeria napoletana che offre anche paitti tipici della cucina
22    partenopea', TRUE, '01-03-2024',
23    'Ottimo', 15, 2, 5);
24
25 INSERT INTO Sconto(Utente)
26 VALUES ('agnese@email.com');
27
28 INSERT INTO
29   Rider(Codice, Mancia, Stato, PosizioneGPS, NumeroConsegneEffettuate, Veicolo,
30   Km)
31  VALUES ('678950', 3.0, 'disponibile', '43.48,9.20', 10, 'monopattino',
32    30.00),
33    ('157893', 1.0, 'occupato', '50.48,6.30', 15, 'bicicletta elettrica',
34    18.00);
35
36 INSERT INTO Recensione(IDRecensione, Autore, Data, Ora, Destinatario, Valutazione)
37 VALUES ('123re', 'mario@email.com', '06-05-2023', '14:05:20', 'Ristorante',
38    3),
39    ('934oo', 'agnese@email.com', '20-07-2022', '20:05:20', 'Rider', 2);
40
41 INSERT INTO
42   Piatto(Titolo, IDRistorante, ListaIngredienti, ListaAllergeni, Prezzo,
43   Immagine, Sconto)
44  VALUES ('Agnolotti', '789d', 'Farina, uova, carne di maiale', 'Contiene
45    glutine, uovo, latte e frutta a guscio. Può contenere tracce di pesce,
46    soia, molluschi, arachidi, senape, sesamo e sedano.', 'Ristorante',
47    13.50,
48    'https://wips.plugin.it/cips/buonissimo.org/cms/2012/09/Agnolotti.jpg', FALSE),
49    ('Margherita', '456U', 'Farina, salsa di pomodoro, mozzarella, olio
50    evo, basilico', 'Contiene glutine, latticini', 10.00,
51    'https://it.ooni.com/cdn/shop/articles/Margherita-9920.jpg', TRUE);
52
53 INSERT INTO Ordine(Data, Ora, StatoOrdine, Richiedente)
54 VALUES ('06-05-2023', '14:00:00', 'Consegnato', 'mario@email.com'),
55    ('20-11-2022', '10:05:00', 'Annullato', 'agnese@email.com');
56
57 INSERT INTO Commento(IDCommento, Data, IDRecensione, Autore, Corpo)
58 VALUES ('dhg7-fo9', '20-07-2022', '934oo', 'agnese@email.com', 'Rider veloce
59    e puntuale.');
```

```
47 INSERT INTO Chat(Messaggi, Destinatario, Cliente)
48 VALUES ('Avrei dei problemi con un ordine effettuato', 'Ristorante',
          'mario@email.com');
49
50 INSERT INTO Categoria(Nome, IDRistorante)
51 VALUES ('Cucina Tipica', '789d'),
52          ('Pizza', '456U')
```