

Progetto di Base di Dati

Andreea Cojocaru, Veronica Bosso

Anno accademico 2023/2024

Indice

1	Progettazione concettuale	2
1.1	Requisiti iniziali	2
1.2	Glossario dei termini	4
1.3	Requisiti rivisti e strutturati in gruppi di frasi omogenee	5
1.4	Schema E-R principale + <i>business rules</i>	7
2	Progettazione logica	9
2.1	Tavola dei volumi	9
2.2	Tavola delle operazioni	10
2.3	Ristrutturazione dello schema E-R	11
2.3.1	Analisi delle ridondanze	11
2.3.2	Eliminazione delle generalizzazioni	14
2.3.3	Eliminazione degli attributi composti e multivalore	16
2.3.4	Scelta degli identificatori principali	17
2.4	Schema E-R ristrutturato + <i>business rules</i>	18
2.5	Schema relazionale + vincoli di integrità referenziale	20
3	Implementazione	21
3.1	DDL di creazione del database	21
3.2	DML di popolamento di tutte le tabelle del database	25
3.3	Operazioni	29

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 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 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 stelline aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al tipo di cibo offerto (ad esempio: fast food, vegetariano, ...). I ristoranti che dimostrano di saper garantire un ottimo servizio (almeno 20 ordini consegnati correttamente, una valutazione clienti maggiore o uguale a 4.5 stelline su cinque, una percentuale massima di ordini annullati dal ristorante dell'1.5%, una percentuale massima di ordini con reclami del 2.5%) sono considerati Top Partner. I Top Partner compaiono in sezioni dedicate all'interno dell'app mobile Cibora e ricevono uno speciale badge che attesta il loro servizio eccellente, aiutando ad aumentare la credibilità e ottenere la fiducia dei clienti. Per i Top Partner si vuole tenere traccia della data in cui sono entrati a far parte della categoria.

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

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

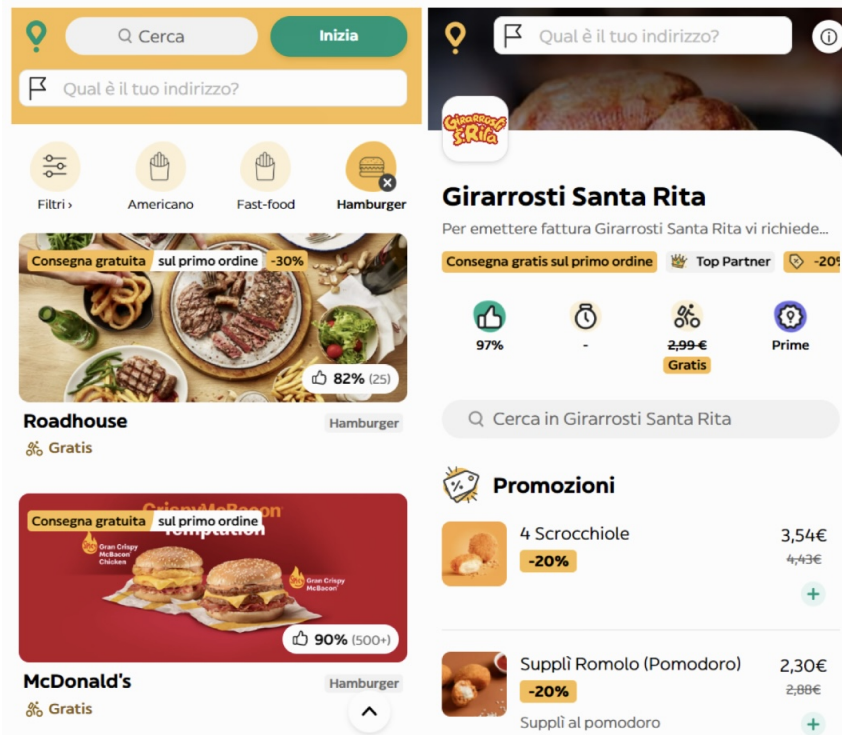


Figure 1: Figura 1 (a) La lista dei ristoranti con filtro “Hamburger”. (b) I dettagli di un ristorante.

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

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

Dopo che l'ordine è stato effettuato l'utente ha la possibilità di chattare sia con il ristorante 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	Sinonimi	Breve descrizione	Termini collegati
Utenti	Clienti	Persona registrata su <i>Cibora</i> che effettua ordini e gestisce il borsellino	Utenti premium, Ordine, Borsellino elettronico
Utenti premium	Modalità premium	Modalità di abbonamento per utenti che garantisce priorità sugli ordini.	Utente
Borsellino elettronico		Sistema di pagamento interno alla piattaforma <i>Cibora</i> , ricaricabile tramite vari metodi di pagamento.	Utenti, Ordine
Codice sconto		Codice che offre sconti sull'importo totale dell'ordine, ottenibile con un certo numero di ordini completati.	Utente, Ordine
Ristoranti	Ristoratori	Attività registrata su <i>Cibora</i> che offre piatti per la consegna. Può appartenere a una o più categorie.	Menù, Top Partner, Recensione
Top Partner		Ristorante che garantisce un ottimo servizio	Ristorante
Piatti	Portata, Pietanza	Offerta dei ristoranti, con ingredienti, prezzo e categoria	Ordine, Ristorante
Menu	Lista di piatti da ordinare	Lista di piatti proposti da un ristorante	Ristorante, Piatti
Rider	Fattorini, Ciclofattorini	Persona responsabile delle consegne	Ordine, Recensione
Ordine	Ordinazione, Cibo, Consegne	Lista di piatti ordinata da un utente per la consegna	Utente, Piatti, Rider
Chat	Chattare	Funzione che consente agli utenti di contattare il ristorante o il rider per problemi relativi all'ordine (ritardo, mancata consegna, ecc.).	Utente, Ristorante, Rider
Recensione	Valutazione	Feedback che un utente può lasciare per un ristorante o un rider; include un commento facoltativo.	Utente, Ristorante, Rider
Classifica		Lista aggiornata mensilmente su più elementi (es. riders più veloci, cibi più popolari, ristoranti con più recensioni positive, ecc.).	Rider, Piatto, Ristorante, Utente

Table 1: Glossario dei termini usati nel servizio *Cibora*

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

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 **fattorini** che effettuano le consegne.

- Utenti
 - Ogni utente deve registrarsi inserendo
 - * nome
 - * email
 - * password
 - * numero di telefono
 - * indirizzo.
 - È possibile sottoscrivere la modalità **premium** che garantisce priorità sugli ordini.
 - Ogni utente può ottenere **codici sconto** da utilizzare per gli ordini in base al numero di ordini passati.
 - Ogni utente può ispezionare **ordini passati** nel proprio profilo e inviare eventuali **reclami** ai ristoranti.
- Borsellino elettronico
 - Ogni utente ha un **borsellino elettronico**, che deve ricaricare tramite metodo di pagamento inserito dall'utente, per poter effettuare ordini.
Il saldo del borsellino è aggiornato a ogni ordine e può essere incrementato in qualsiasi momento.
- Ristoranti
 - Ogni ristorante è identificato da
 - * nome
 - * descrizione
 - * indirizzo
 - * un costo di spedizione
 - * un'immagine di profilo
 - * una valutazione in stelle aggiornata in base alle recensioni ricevute nell'ultima settimana.
 - Ogni ristorante appartiene a una o più **categorie** in base al tipo di cibo offerto
 - Ogni ristorante che soddisfa un particolare criterio ottiene lo status di **Top Partner**.
Criterio:
 - * almeno 20 ordini consegnati correttamente
 - * valutazione minima di 4.5 stelle su 5
 - * percentuale massima di ordini annullati pari a 1.5
 - * percentuale massima di reclami pari a 2.5
- Piatti
 - Ogni piatto ha
 - * un titolo,
 - * un'immagine,
 - * una lista di ingredienti,

- * una lista di allergeni,
 - * un prezzo,
 - * un eventuale sconto.
- Ogni piatto è organizzato in **menù** per categorizzare i piatti proposti dai ristoranti
- Ogni piatto appartiene ad uno o più **menù**
- Ordini
 - Ogni ordine **effettuato** è composto dai piatti selezionati dall'utente e associati ad un ristorante.
 - Ogni ordine può essere **annullato** sia dagli utenti sia dai ristoratori, fino a quando non vengono affidati a un rider per la consegna.
 - Dopo aver effettuato un ordine, gli utenti possono **chattare** con il ristorante e con il rider per risolvere eventuali problemi.
 - Al termine dell'ordine, l'utente ha la possibilità di **recensire** ristorante e rider, assegnando un punteggio da 1 a 5 stelle e aggiungendo un commento testuale facoltativo.
 - L'utente ha inoltre la possibilità di lasciare una **mancia** al rider per il servizio di consegna.
- Riders
 - Ogni rider è identificato da un
 - * codice unico
 - * uno stato che può essere
 - *occupato*
 - *disponibile*
 - *fuori servizio*
 - Ogni rider è classificato per **mezzo di trasporto**: bicicletta normale, bicicletta elettrica o monopattino.
 - * I riders con monopattino devono dichiarare la **distanza massima** percorribile prima che la batteria si scarichi.
 - Al momento della conferma dell'ordine, il sistema trova il rider libero che minimizza la somma della distanza tra ristorante e cliente. Se la distanza supera i 10 km, il sistema considera soltanto i riders con bici elettrica.
 - Per ogni rider, si registra
 - * il numero di consegne effettuate,
 - * l'ora in cui l'ordine viene affidato a un rider
 - * l'ora di quando è stato recapitato al cliente.
- Classifiche
 - Ogni mese la piattaforma aggiorna alcune **classifiche** basate sulle prestazioni e sui dati di utilizzo della piattaforma:
 - * Riders più veloci nelle consegne
 - * Cibi più popolari
 - * Ristoranti con il maggior numero di recensioni positive
 - * Clienti con la spesa più elevata

1.4 Schema E-R principale + *business rules*

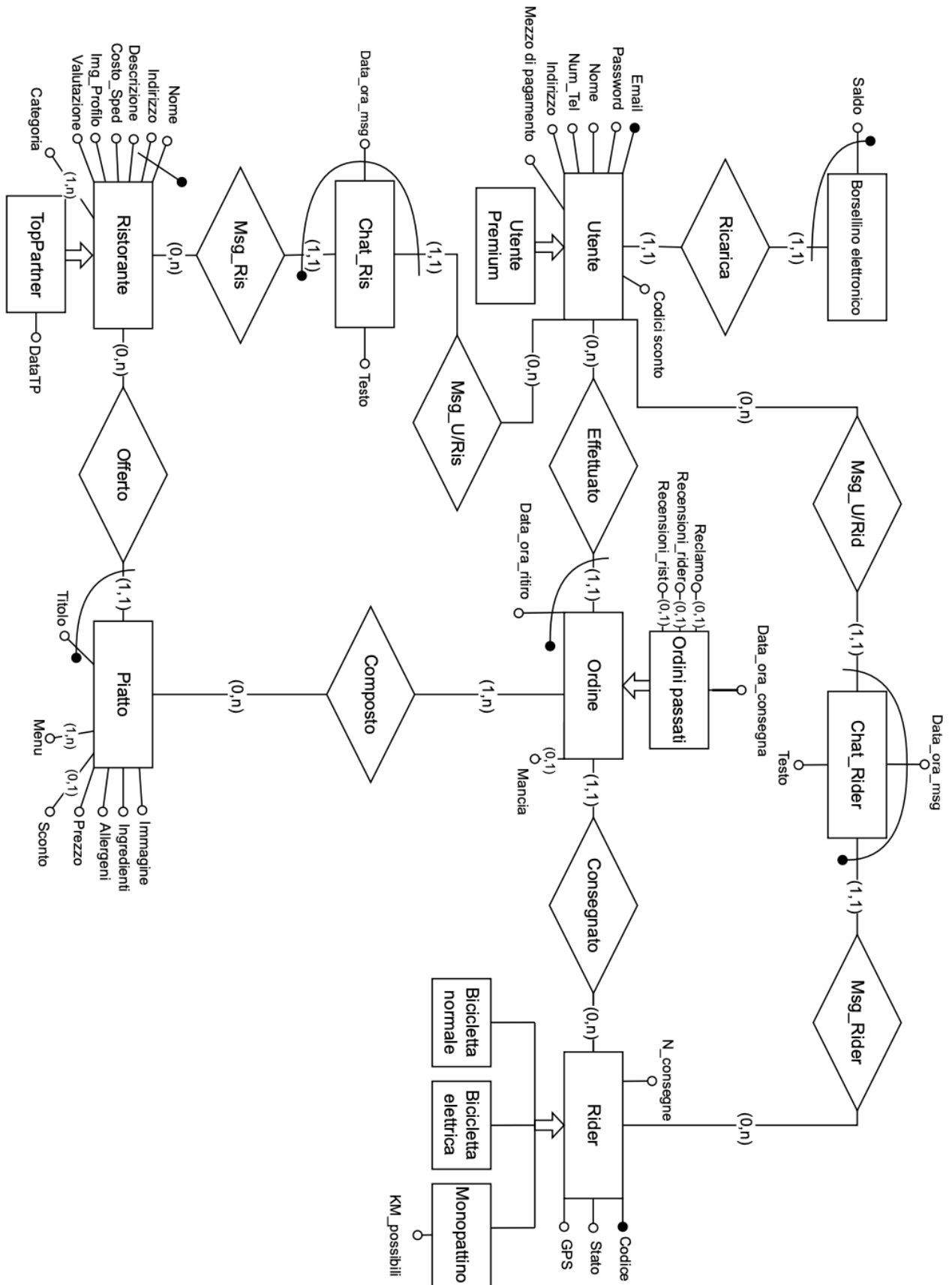


Figure 2: Schema ER

Business Rules

- Un utente deve essere iscritto per usufruire del servizio.
 - Gli utenti premium ricevono priorità sugli ordini.
 - Un utente colleziona codici sconto in base agli ordini passati.
 - Gli ordini possono essere annullati da utenti o ristoranti finché non sono affidati a un rider per la consegna; un ordine annullato prima del ritiro da parte del rider non viene salvato nel database.
 - Le chat tra utente-ristorante e utente-rider possono essere usate soltanto dopo che il rider ha ritirato l'ordine dal ristorante.
 - Il numero di stellette (valutazione) è aggiornato ogni lunedì sulla base delle percentuali di recensioni positive dell'ultima settimana.
 - Un ristorante entra nella categoria Top Partner se rispetta tutti i seguenti parametri contemporaneamente:
 - Almeno 20 ordini consegnati correttamente;
 - Valutazione clienti maggiore o uguale a 4.5 stelline su cinque;
 - Percentuale massima di ordini annullati dai ristoranti dell'1.5%;
 - Percentuale massima di ordini con reclami del 2.5%;
- I ristoranti che diventano Top Partner ricevono un badge di attestazione.
- La consegna dell'ordine viene effettuata dal rider libero con la somma minima della distanza dal ristorante più la distanza dall'utente.
 - Per ordini che prevedano un tragitto "posizione corrente del rider-> ristorante -> cliente" superiore ai 10 km, solo i rider con bici elettrica vengono interpellati.
 - 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.

2 Progettazione logica

2.1 Tavola dei volumi

Concetto	Tipo	Volume	Motivazioni scelta
Utente	E	1.200.000	Stima basata sul numero di utenti di servizi simili a livello nazionale.
Utente Premium	E	180.000	Si stima una percentuale del 15% di utenti premium.
Borsellino	E	1.200.000	Ogni utente registrato ha un borsellino.
Ristorante	E	50.000	Stima basata sul numero di ristoranti affiliati a servizi simili a livello nazionale.
Top Partner	E	10.000	Si stima una percentuale del 20% di Top Partner sul totale dei ristoranti.
Piatto	E	750.000	Totale dei piatti offerti da Cibora, stimando 15 piatti per ristorante
Ordine	E	24.000.000	Stima basata sul numero di ordini su servizi simili, a livello nazionale annualmente.
Ordini passati	E	24.000.000	Stima al pari degli ordini
Rider	E	100.000	Stima basata sul numero di rider di servizi simili a livello nazionale.
Bicicletta normale	E	15.000	Si stima che il 15% dei rider utilizzi una bicicletta normale
Bicicletta elettrica	E	45.000	Si stima che il 45% dei rider utilizzi una bicicletta elettrica
Monopattino	E	40.000	Si stima che il 40% dei rider utilizzi un monopattino
Ricarica	A	20.000.000	Ogni volta che il borsellino ha saldo insufficiente a ricoprire il totale dell'ordine
Effettuato	A	24.000.000	Stima al pari degli ordini
Offerto	A	750.000	Stima al pari degli piatti
Consegnato	A	24.000.000	Stima al pari degli ordini
Composto	A	72.000.000	Si stima 3 piatti per ordine
ChatRider	A	2.400.000	Si stima che 1 volta su 10 utente inizi una chat con il rider
ChatRist	A	2.400.000	Si stima che 1 volta su 10 utente inizi una chat con il ristorante

Table 2: Tavola dei volumi

2.2 Tavola delle operazioni

N	Operazioni	Descrizione	Tipo	Frequenza
1	Registrazione utente	Un utente si registra sulla piattaforma	I	Circa 100 utenti al giorno
2	Creazione ordine	Un utente effettua un nuovo ordine	I	Circa 65.750* volte al giorno
3	Assegnazione rider	Un ordine viene affidato a un rider	B	Circa 65.750* volte al giorno
4	Ricarica borsellino	Un utente ricarica il borsellino	I	Circa 55.000 volte al giorno
5	Recensione ristorante e rider	Un utente recensisce ristorante e rider	I	Si stima che 1 ordine su 1.000, un utente recensisce
6	Aggiornamento mensile delle classifiche	Calcolo dei riders più veloci, cibi più popolari, ristoranti con più recensioni positivi, clienti che hanno speso di più	B	1 volta al mese
7	Assegnazione badge Top Partner	Promozione da Ristorante a Top Partner	B	Ogni volta che un ristorante soddisfa i requisiti
8	Ricerca piatti da ordinare	Un utente che vuole ordinare prima cerca su Cibora ciò che vuole	I	Circa 70.000 volte al giorno
9	Aggiornamento valutazione ristorante	Il numero di stelline di un ristorante viene aggiornato sulla base delle recensioni positive ricevute	B	Ogni Lunedì
10	Ispezionare ordini passati	Un utente può visualizzare gli ordini passati	I	Circa 500 volte al giorno
11	Calcolo consegne rider	Per ogni rider è possibile vedere il numero di consegne effettuate	B	Circa 250 volte al giorno

*stima ordini giornalieri (24.000.000/365)

Table 3: Tavola delle operazioni

2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

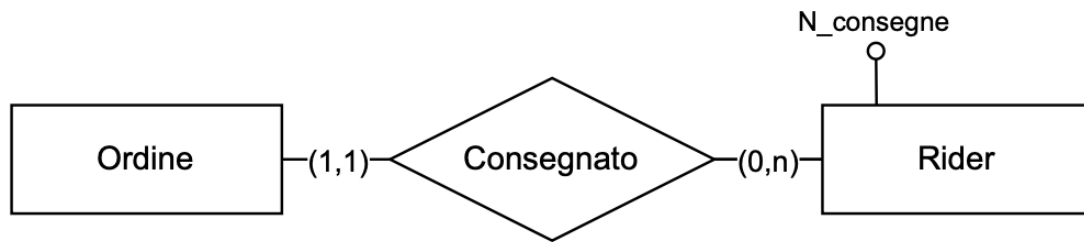


Figure 3: Ridondanza

Analisi Operazione 11

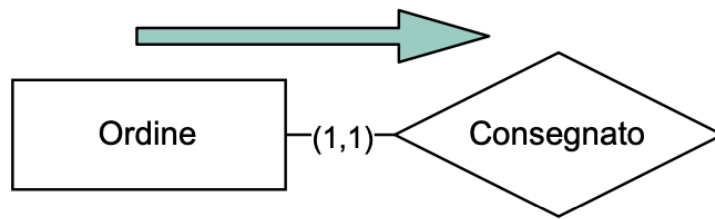
Ridondanza da attributi derivabili dal conteggio di occorrenze di attributi di altre entità. Il numero di consegne effettuate da un rider è ottenibile contando le occorrenze dell'associazione Consegnato a cui il rider partecipa.

Concetto	Tipo	Volume
Rider	E	100.000
Ordine	E	24.000.000
Consegnato	A	24.000.000

Table 4: Tavola dei volumi

Operazione	Descrizione	Tipo	Frequenza
1	Un rider consegna un ordine effettuato da un utente	I	65.750 al giorno
2	Visualizzazione del numero di ordini consegnati di un rider	B	250 al giorno

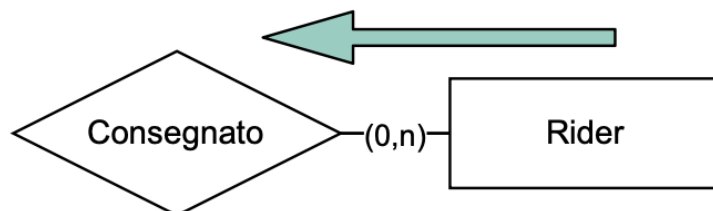
Table 5: Tavola delle operazioni



Analisi senza ridondanza: Operazione 1

Concetto	Costrutto	Accessi	Tipo
Ordine	Entità	1	S
Consegnato	Associazione	1	S

Table 6: Tavola degli accessi



Analisi senza ridondanza: Operazione 2

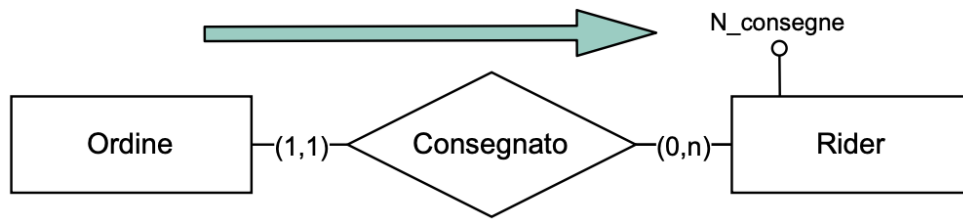
Concetto	Costrutto	Accessi	Tipo
Rider	Entità	1	L
Consegnato	Associazione	240*	L

*Numero di ordini medio consegnati da un rider = 24.000.000/100.000

Table 7: Tavola degli accessi

Costi:

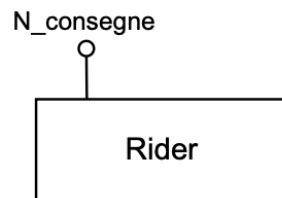
- Spazio: 0 byte
- Tempo:
 - Operazione 1: $2 * 65.750$ accessi in scrittura^(vale doppio) al giorno
 - Operazione 2: $240 * 250$ accessi in lettura al giorno
 - Totale accessi: $(131.500 * 2) + 60.000 = 323.000$ accessi al giorno



Analisi con ridondanza: Operazione 1

Concetto	Costrutto	Accessi	Tipo
Rider	Entità	1	L
Rider	Entità	1	S
Ordine	Entità	1	S
Consegnato	Associazione	1	S

Table 8: Tavola degli accessi



Analisi con ridondanza: Operazione 2

Concetto	Costrutto	Accessi	Tipo
Rider	Entità	1	L

Table 9: Tavola degli accessi

Costi:

- Spazio: assumendo di usare 4 byte per memorizzare il numero di ordini per ogni rider, $4 * 100.000 \text{ byte} \approx 2^{19}$
- Tempo:
 - Operazione 1: $3 * 65.750$ accessi in scrittura e 65.750 accessi in lettura al giorno
 - Operazione 2: 250 accessi in lettura
 - Totale accessi: $[(65.750 * 3) * 2] + 65.750 + 250 = 460.500$ accessi al giorno

Conclusioni finali:

La soluzione con ridondanza occupa uno spazio di quasi un Mbyte e prevede circa 100.000 accessi in più rispetto all'opzione senza ridondanza.

Considerando questi dati, abbiamo deciso di non mantenere la ridondanza.

2.3.2 Eliminazione delle generalizzazioni

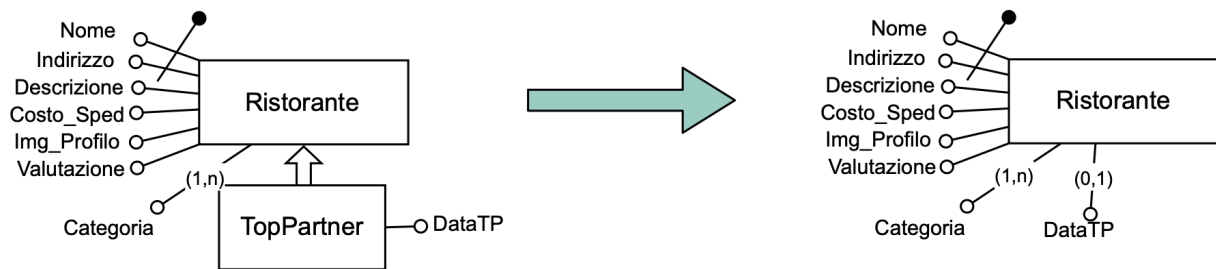


Figure 4: Generalizzazione *Top Partner*

- **Generalizzazione *Top Partner***: accorpamento del figlio della generalizzazione nel genitore. L'entità figlia Top Partner viene eliminata, il suo attributo viene aggiunto all'entità genitore Ristorante. Questo attributo verrà popolato con la data nella quale il ristorante è diventato Top Partner; se il ristorante non ha mai raggiunto questo titolo l'attributo rimarrà nullo.

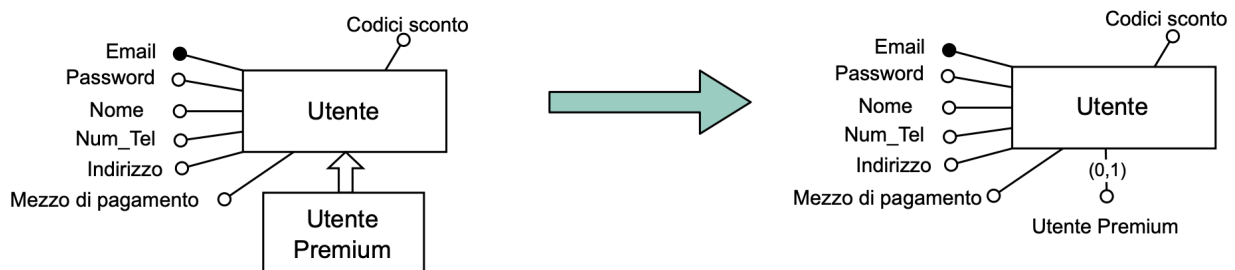


Figure 5: Generalizzazione *Utente Premium*

- **Generalizzazione *Utente Premium***: accorpamento del figlio della generalizzazione nel genitore. L'entità figlia Utente Premium viene eliminata. All'entità genitore Utente viene aggiunto un attributo Utente Premium. Questo attributo conterrà "true" se l'utente in questione ha un abbonamento Premium, altrimenti rimarrà nullo.

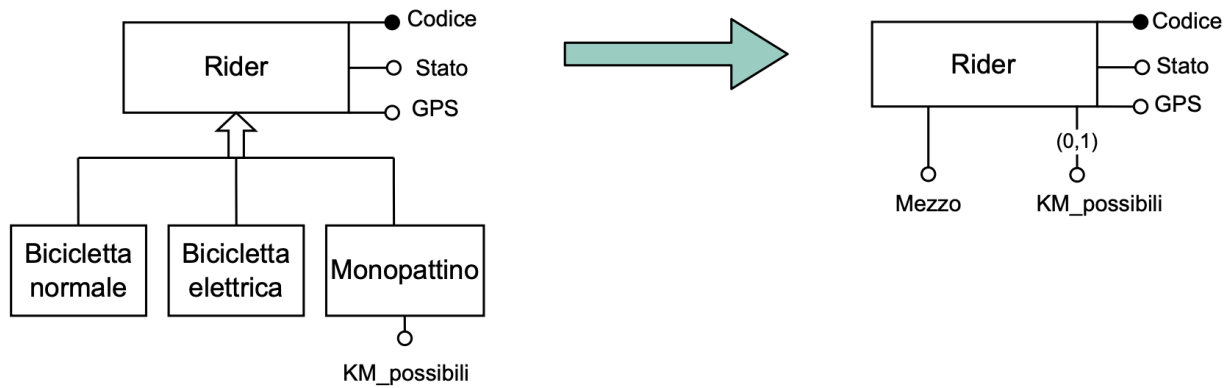


Figure 6: Generalizzazione *Rider*

- **Generalizzazione *Rider*:** accorpamento dei figli della generalizzazione nel genitore. L'entità figlie Bicicletta normale, Bicicletta elettrica, Monopattino vengono eliminate. All'entità genitore Rider vengono aggiunti due attributi, Mezzo e KM_possibili. Mezzo indica il mezzo di spostamento del rider, avrà uno dei seguenti valori: Bicicletta normale, Bicicletta Elettrica o Monopattino. KM_possibili indica il numero di chilometri che un monopattino può percorrere prima che si scarichi la batteria. Quest'ultimo attributo sarà nullo per tutti i rider che non usano un monopattino.

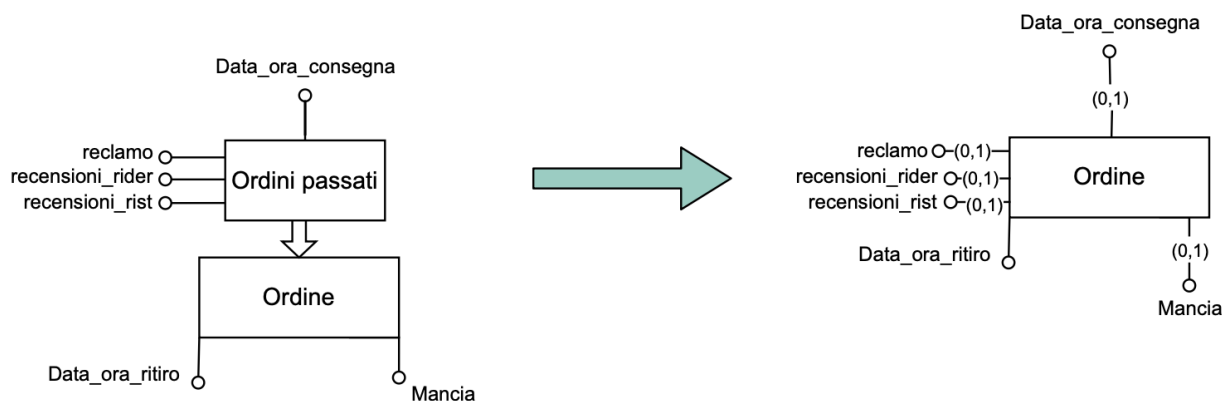


Figure 7: Generalizzazione *Ordini passati*

- **Generalizzazione *Ordini passati*:** accorpamento del figlio della generalizzazione nel genitore. L'entità figlia Ordini passati viene eliminata. All'entità genitore Utente vengono aggiunti tutti gli attributi dell'entità figlia. Tutti questi attributi saranno facoltativi, quindi possono contenere dei dati oppure essere nulli in base alla presenza o meno di reclami e/o recensioni.

2.3.3 Eliminazione degli attributi composti e multivalore

Abbiamo deciso di trasformare gli attributi multivalore reificando l'attributo e aggiungendo un'associazione.

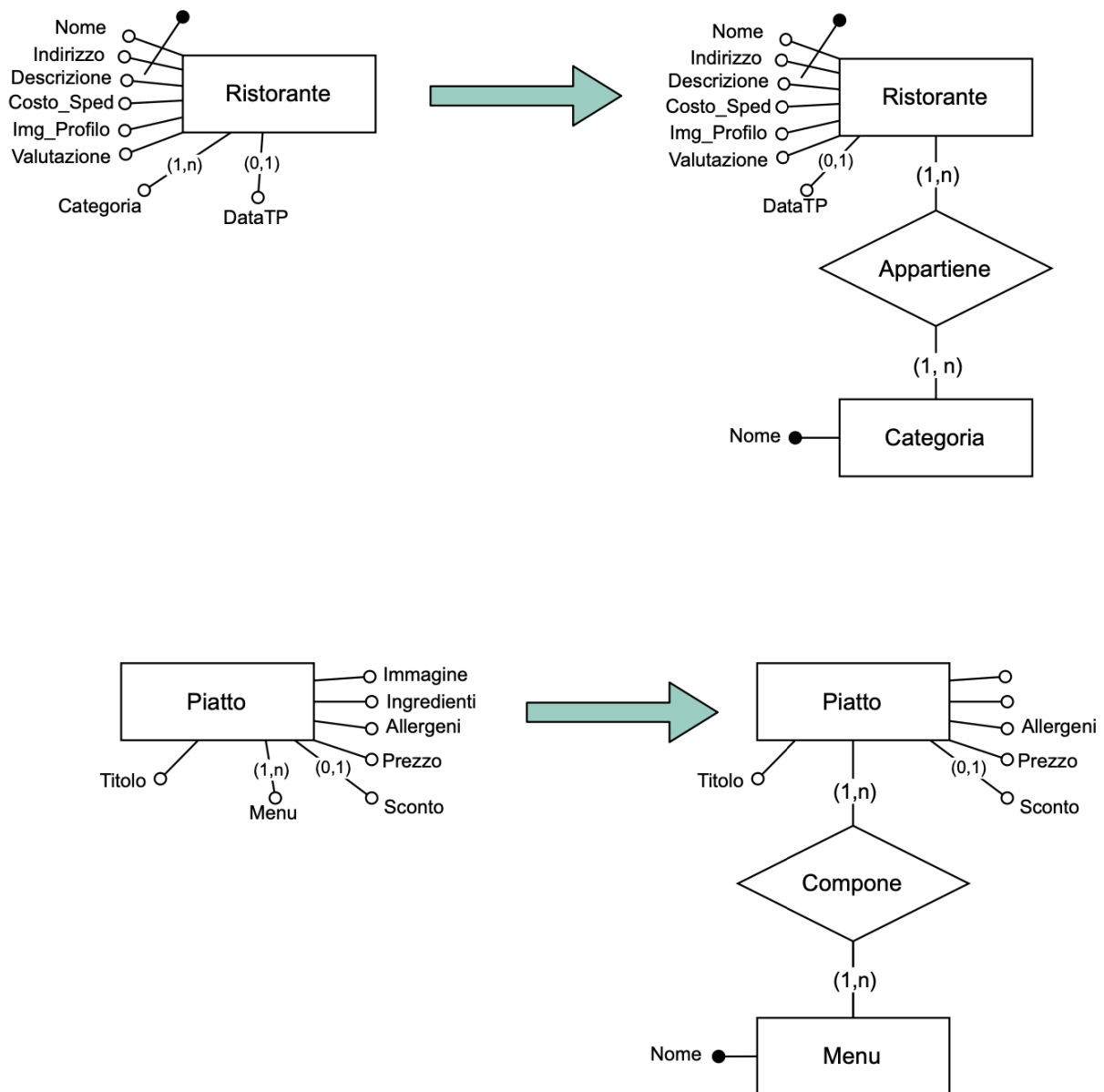


Figure 8: Attributi *multivalore*

2.3.4 Scelta degli identificatori principali

Per motivi di performance, come nel caso di identificatori poco efficienti (ad esempio una stringa troppo lunga), abbiamo creato degli identificatori “surrogati” (ID).

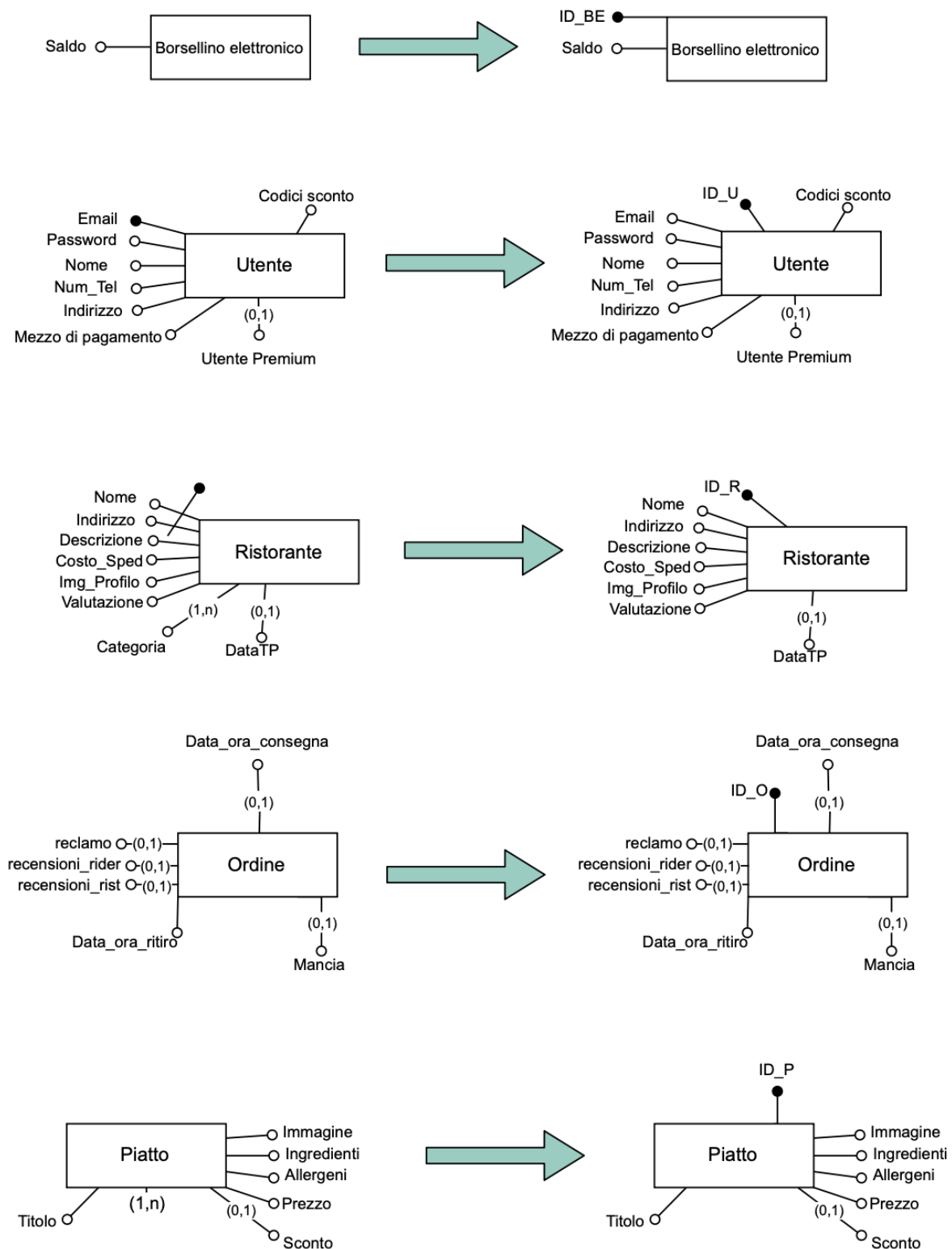


Figure 9: Identificatori

2.4 Schema E-R ristrutturato + *business rules*

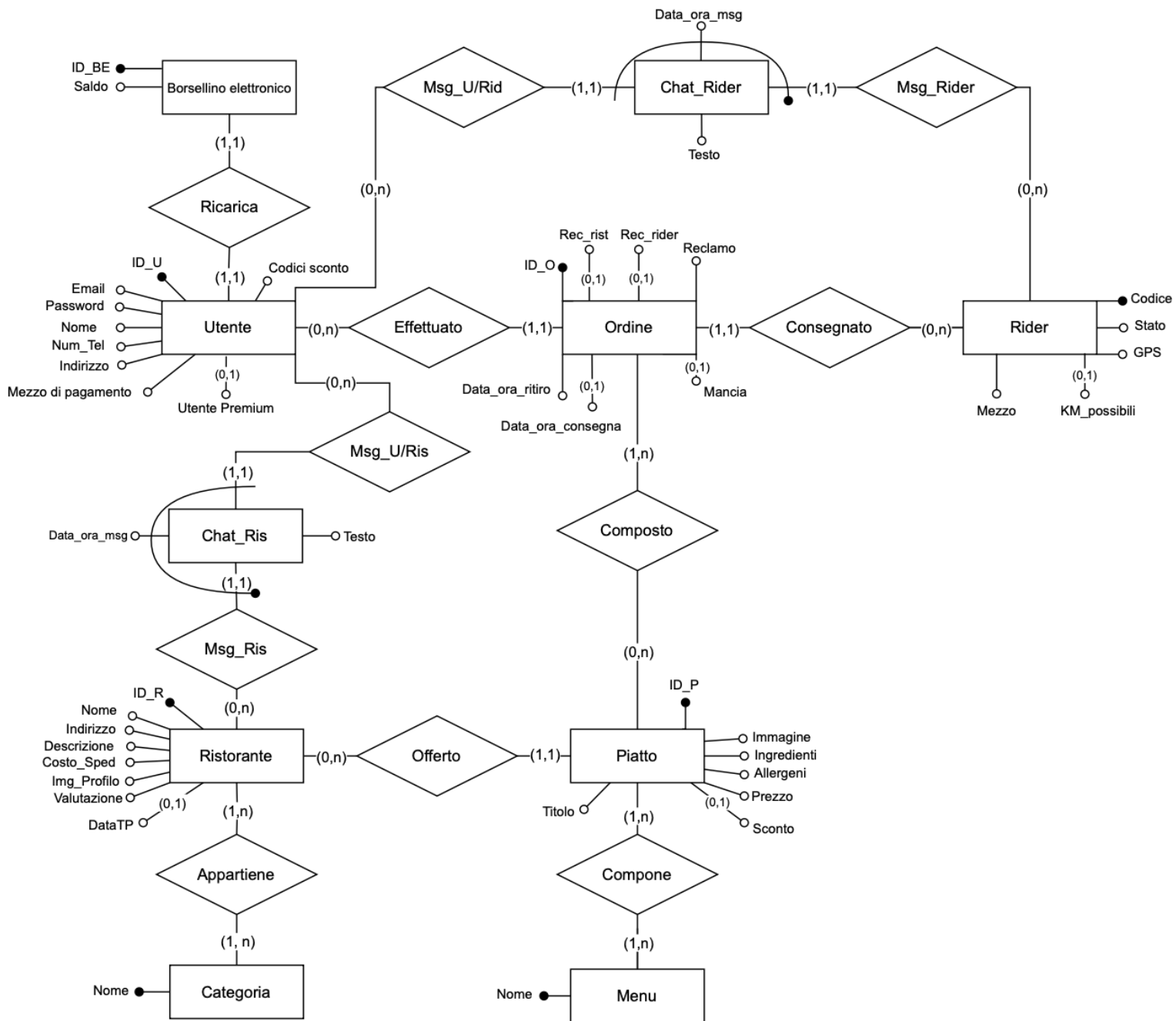


Figure 10: Schema ER ristrutturato

Business Rules

- Un utente deve essere iscritto per usufruire del servizio.
 - Gli utenti premium ricevono priorità sugli ordini.
 - Un utente colleziona codici sconto in base agli ordini passati.
 - Gli ordini possono essere annullati da utenti o ristoranti finché non sono affidati a un rider per la consegna; un ordine annullato prima del ritiro da parte del rider non viene salvato nel database.
 - Le chat tra utente-ristorante e utente-rider possono essere usate soltanto dopo che il rider ha ritirato l'ordine dal ristorante.
 - Il numero di stellette (valutazione) è aggiornato ogni lunedì sulla base delle percentuali di recensioni positive dell'ultima settimana.
 - Un ristorante entra nella categoria Top Partner se rispetta tutti i seguenti parametri contemporaneamente:
 - Almeno 20 ordini consegnati correttamente;
 - Valutazione clienti maggiore o uguale a 4.5 stelline su cinque;
 - Percentuale massima di ordini annullati dai ristoranti dell'1.5%;
 - Percentuale massima di ordini con reclami del 2.5%;
- I ristoranti che diventano Top Partner ricevono un badge di attestazione.
- La consegna dell'ordine viene effettuata dal rider libero con la somma minima della distanza dal ristorante più la distanza dall'utente.
 - Per ordini che prevedano un tragitto "posizione corrente del rider-> ristorante -> cliente" superiore ai 10 km, solo i rider con bici elettrica vengono interpellati.
 - 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.

2.5 Schema relazionale + vincoli di integrità referenziale

Utente (ID_U, Email, Password, Nome, Num_Tel, Indirizzo, Mezzo di pagamento, Utente Premium, Codici_Sconto, ID_BE)

- Utente(ID_BE) referencia Borsellino(ID_BE).
- Utente(Email) unique

Borsellino Elettronico(ID_BE, Saldo)

Ordine(ID_O, Rec_rist, Rec_rider, Reclamo, Mancias, Data_ora_ritiro, Data_ora_consegna, ID_U, Codice)

- Ordine(ID_U) referencia Utente(ID_U)
- Ordine(Codice) referencia Rider(Codice)

Rider(Codice, Stato, GPS, Mezzo, KM_possibili)

Composto(ID_O, ID_P)

- Composto(ID_O) referencia Orine(ID_O)
- Composto(ID_P) referencia Piatto(ID_P)

Piatto(ID_P, Immagine, Ingredienti, Allergeni, Prezzo, Sconto, Titolo, ID_R)

- Piatto(ID_R) referencia Ristorante(ID_R)

Menu(Nome)

Compone(ID_P, Menu)

- Compone(ID_P) referencia Piatto(ID_P)
- Compone(Menu) referencia Menu(Nome)

Ristorante(ID_R, Nome, Indirizzo, Descrizione, Costo_Sped, Img_Profilo, Valutazione, DataTP)

Categoria(Nome)

Appartiene(ID_R, Categoria)

- Appartiene(ID_R) referencia Ristorante(ID_R)
- Appartiene(Categoria) referencia Categoria(Nome)

Chat_Ris(Data_ora_msg, ID_U, ID_R, Testo)

- Chat_Ris(ID_U) referencia Utente(ID_U)
- Chat_Ris(ID_R) referencia Ristorante(ID_R)

Chat_Rider(Data_ora_msg, ID_U, Rider, Testo)

- Chat_Rider(ID_U) referencia Utente(ID_U)
- Chat_Rider(Rider) referencia Rider(Codice)

3 Implementazione

3.1 DDL di creazione del database

Tabella *Borsellino Elettronico*

```
CREATE TABLE BE (  
    ID_BE SERIAL PRIMARY KEY,  
    Saldo DECIMAL(10, 2) NOT NULL  
);
```

Tabella *Utente*

```
CREATE TABLE Utente (  
    ID_U SERIAL PRIMARY KEY,  
    Email VARCHAR(255) UNIQUE NOT NULL,  
    Password VARCHAR(50) NOT NULL,  
    Nome VARCHAR(100) NOT NULL,  
    Num_Tel VARCHAR(10),  
    Indirizzo VARCHAR(255) NOT NULL,  
    Mezzo_di_pagamento VARCHAR(50),  
    Utente_Premium BOOLEAN,  
    Codici_Sconto VARCHAR(3),  
    ID_BE INT,  
    FOREIGN KEY (ID_BE) REFERENCES Borsellino(ID_BE)  
);
```

Tabella *Rider*

```
CREATE TABLE Rider (  
    Codice SERIAL PRIMARY KEY,  
    Stato VARCHAR(20),  
    GPS VARCHAR(50),  
    Mezzo VARCHAR(50),  
    KM_possibili INT  
);
```

Tabella *Ordine*



```
CREATE TABLE Ordine (  
    ID_O INT PRIMARY KEY,  
    Rec_rist TEXT,  
    Rec_rider TEXT,  
    Reclamo TEXT,  
    Mancia DECIMAL(10, 2),  
    Data_ora_ritiro TIMESTAMP,  
    Data_ora_consegna TIMESTAMP,  
    ID_U INT,  
    Codice INT,  
    FOREIGN KEY (ID_U) REFERENCES Utente(ID_U)   
    FOREIGN KEY (Codice) REFERENCES Rider(Codice)   
);
```

Tabella *Ristorante*

```
CREATE TABLE Ristorante (  
    ID_R SERIAL PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Indirizzo VARCHAR(255) NOT NULL,  
    Descrizione TEXT,  
    Costo_Sped DECIMAL(10, 2),  
    Img_Profilo VARCHAR(255),  
    Valutazione DECIMAL(2, 1),  
    DataTP DATE  
);
```

Tabella *Piatto*

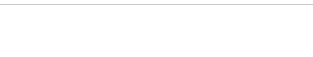
```
CREATE TABLE Piatto (  
    ID_P SERIAL PRIMARY KEY,  
    Immagine VARCHAR(255),  
    Ingredienti TEXT,  
    Allergeni TEXT,  
    Prezzo DECIMAL(10, 2) NOT NULL,  
    Sconto DECIMAL(5, 2),  
    Titolo VARCHAR(100),  
    ID_R INT,  
    FOREIGN KEY (ID_R) REFERENCES Ristorante(ID_R)   
);
```

Tabella *Composto*

```
CREATE TABLE Composto (  
    ID_O INT,  
    ID_P INT,  
    CONSTRAINT pk_Composto PRIMARY KEY (ID_O, ID_P),  
    FOREIGN KEY (ID_O) REFERENCES Ordine(ID_O)  
    FOREIGN KEY (ID_P) REFERENCES Piatto(ID_P)  
);
```

Tabella *Menu*

```
CREATE TABLE Menu (  
    Nome VARCHAR(50) PRIMARY KEY  
);
```

Tabella *Compone*

```
CREATE TABLE Compone (  
    ID_P INT,  
    Menu VARCHAR(50),  
    CONSTRAINT pk_Compone PRIMARY KEY (ID_P, Menu),  
    FOREIGN KEY (ID_P) REFERENCES Piatto(ID_P)  
    FOREIGN KEY (Menu) REFERENCES Menu(Nome)  
);
```

Tabella *Categoria*

```
CREATE TABLE Categoria (  
    Nome VARCHAR(50) PRIMARY KEY  
);
```

Tabella *Appartiene*

```
CREATE TABLE Appartiene (  
    ID_R INT,  
    Categoria VARCHAR(50),  
    CONSTRAINT pk_Appartiene PRIMARY KEY (ID_R, Categoria),  
    FOREIGN KEY (ID_R) REFERENCES Ristorante(ID_R) ,  
    FOREIGN KEY (Categoria) REFERENCES Categoria(Nome)   
);
```

Tabella *Chat_Ris*

```
CREATE TABLE Chat_Ris (  
    Data_ora_msg TIMESTAMP,  
    ID_U INT,  
    ID_R INT,  
    Testo TEXT,  
    CONSTRAINT PK_chat_Ris PRIMARY KEY (Data_ora_msg, ID_U, ID_R),  
    FOREIGN KEY (ID_U) REFERENCES Utente(ID_U) ,  
    FOREIGN KEY (ID_R) REFERENCES Ristorante(ID_R)   
);
```

Tabella *Chat_Rider*

```
CREATE TABLE Chat_Rider (  
    Data_ora_msg TIMESTAMP,  
    ID_U INT,  
    Rider VARCHAR(50),  
    Testo TEXT,  
    CONSTRAINT PK_chat_Rider PRIMARY KEY (Data_ora_msg, ID_U, Rider),  
    FOREIGN KEY (ID_U) REFERENCES Utente(ID_U) ,  
    FOREIGN KEY (Rider) REFERENCES Rider(Codice)   
);
```

3.2 DML di popolamento di tutte le tabelle del database

Tabella *Borsellino elettronico*

```
INSERT INTO BE (Saldo)
VALUES (30.20),
       (5.98),
       (14.10),
       (23.56),
       (28.48);
```

Tabella *Utente*

```
INSERT INTO Utente (Email, Password, Nome, Num_Tel, Indirizzo, Mezzo_di_pagamento,
                   Utente_Premium, Codici_Sconto, ID_BE)
VALUES ('amy.pond@gmail.com', 'psw123', 'Amelia Pond', '3124567890',
       'via Roma 2, Milano', 'Satispay', TRUE, '2%', 1),
       ('rory.gilmore@gmail.com', 'psw456', 'Lorelai Gilmore', '3245678971',
       'via Milano 3, Torino', 'Bancomat', FALSE, NULL, 2),
       ('nick.andersen@gmail.com', 'psw789', 'Nicholas Andersen', '3245678901',
       'via Torino 2, Genova', 'Carta di credito', TRUE, '5%', 3),
       ('mario.rossi@gmail.com', 'psw012', 'Mario Rossi', '3345678912',
       'via Genova 10, Bari', 'Carta di debito', FALSE, '3%', 4),
       ('andrea.bruno@gmail.com', 'psw234', 'Andrea Bruno', '3245678903',
       'via Bari 7, Udine', 'Bancomat', FALSE, NULL, 5);
```

Tabella *Rider*

```
INSERT INTO Rider (Stato, GPS, Mezzo, KM_possibili)
VALUES ('Disponibile', '45.5499775, 9.0787175', 'Bicicletta elettrica', NULL),
       ('Disponibile', '45.0736869, 7.6815302', 'Bicicletta normale', NULL),
       ('Non disponibile', '44.3348549, 9.2128674', 'Monopattino', 15),
       ('Non disponibile', '41.1643794, 16.7479558', 'Bicicletta elettrica', NULL),
       ('Occupato', '46.0637, 13.24458', 'Monopattino', 12);
```

Tabella *Ordine*

```
INSERT INTO Ordine (Rec_rist, Rec_rider, Reclamo, Mancia,
                   Data_ora_ritiro, Data_ora_consegna, ID_U, Codice)
VALUES (NULL, NULL, NULL, NULL,
        '2024-11-30 12:34:45', '2024-11-30 12:54:01', 1, 1),
('3 Stelle', '4 Stelle', NULL, 1,
        '2024-11-28 20:17:12', '2024-11-28 20:27:43', 2, 2),
(NULL, NULL, NULL, 2.5,
        '2024-11-30 19:47:21', '2024-11-30 20:01:23', 3, 3)
('5 Stelle, cibo squisito', NULL, NULL, NULL,
        '2024-11-29 14:02:36', '2024-11-29 14:29:52', 4, 4);
('4 Stelle', '5 Stelle, ordine arrivato in orario', NULL, 5,
        '2024-11-30 16:27:39', '2024-11-30 16:33:01', 5, 5);
```

Tabella *Ristorante*

```
INSERT INTO Ristorante (Nome, Indirizzo, Descrizione,
                        Costo_Sped, Img_Profilo, Valutazione, DataTP)
VALUES ('Bella Napoli', '"piazza San Marco 2, Milano"', 'Pizzeria tradizionale',
        2.99, 'piz_BNP.png', 4.7, '2024-11-13'),
('Sonamu', 'corso Peschiera 160, Torino', 'Ristorante coreano',
        2.39, 'soju.png', 4.6, '2024-11-29'),
('Vaggie', 'via XX Settembre 37, Genova', 'Ristorante vegetariano, vegano',
        3.49, 'carota.png', 3.5, NULL),
('Acasa', 'via Don Bosco 4, Bari', 'Ristorante tradizionale rumeno',
        1.49, 'acasa.png', 4.0, NULL),
('Tokyo Sushi', 'via Pradamano 70, Udine', 'All you can eat',
        1.79, 'sashimi.png', 3.9, NULL);
```

Tabella *Piatto*

```
INSERT INTO Piatto (ID_P, Immagine, Ingredienti, Allergeni,
                   Prezzo, Sconto, Titolo, ID_R)
VALUES ('margherita.png', 'sugo, mozzarella, basilico',
        'latticini, glutine', 7.99, NULL, 'Margherita', 1),
('jajaja.png', 'spaghetti coreani, pasta di fagioli dolce, manzo, cetrioli',
        'glutine, soia', 9.5, NULL, 'Jjajangmyeon', 2),
('tofu.png', 'tofu, lattuga, pomodorini, edamame, zucchine, senape al melie',
        'senape', 10, 0.5, 'Tofu Salad', 3),
('sarlmale.png', 'verza, vitello, riso, cipolla, carote', 'glutine',
        10, 2, 'Sarmale', 4),
('onigiri.png', 'riso, tonno, maionese, alga nori', 'uova, pesce, glutine',
        6, NULL, 'Onigiri', 5);
('sashimiS.png', 'salmone, salsa di soia, wasabi, alghe', 'pesce, soia',
        12, NULL, 'Sashimi Salmone', 5);
```

Tabella *Composto*

```
INSERT INTO Composto (ID_O, ID_P)
VALUES  (1,1),
        (2,2),
        (3,3),
        (4,4),
        (5,5), (5,6);
```

Tabella *Menu*

```
INSERT INTO Menu (Nome)
VALUES  ('Menu del giorno'),
        ('Pizze speciali'),
        ('Menu vegano'),
        ('Menu per catering'),
        ('Menu dolci'),
        ('Menu per bambini');
```

Tabella *Compone*

```
INSERT INTO Compone (ID_P, Menu)
VALUES  (4, 'Menu del giorno'), (2, 'Menu del giorno'),
        (1, 'Pizze speciali'),
        (3, 'Menu vegano'),
        (5, 'Menu per catering'), (6, 'Menu per catering'),
        (1, 'Menu per bambini');
```

Tabella *Categoria*

```
INSERT INTO Categoria (Nome)
VALUES  ('Pizzeria'),
        ('Asian fusion'),
        ('Vegano'),
        ('Vegetariano'),
        ('Cucina internazionale'),
        ('Cucina italiana');
```

Tabella *Appartiene*

```
INSERT INTO Appartiene (ID_R, Categoria)
VALUES  (1,'Pizzeria'),
        (2,'Asian fusion'),
        (5,'Asian fusion'),
        (3,'Vegano'),
        (3,'Vegetariano'),
        (4,'Cucina internazionale'),
        (1,'Cucina italiana');
```

Tabella *Chat_Ristorante*

```
INSERT INTO Chat_Ris (Data_ora_msg, ID_U, ID_R, Testo)
VALUES  ('2024-11-30 12:28:10', 1, 1,
        'Potete mettere le posate, per favore?'),
        ('2024-11-30 12:30:23', 1, 1, 'Va bene'),
        ('2024-11-30 12:31:12', 1, 1, 'Grazie mille'),
        ('2024-11-30 19:40:17', 3, 3,
        'Potete mettere fagioli al posto del edamame?'),
        ('2024-11-30 19:43:34', 3, 3,
        'Non abbiamo i fagioli, vanno bene i ceci?'),
        ('2024-11-30 19:44:47', 3, 3,
        'Si, va bene, grazie mille');
```

Tabella *Chat_Rider*

```
INSERT INTO Chat_Rider (Data_ora_msg, ID_U, Rider, Testo)
VALUES  ('2024-11-28 20:20:20', 2, 2,
        'Può non suonare il campanello,
        mi chiami a questo numero XXX-XXX-XXXX'),
        ('2024-11-28 20:21:45', 2, 2, 'ok'),
        ('2024-11-28 20:22:00', 2, 2, 'Grazie mille');
```

3.3 Operazioni

Modifica

Modifica saldo del *borsellino*

```
UPDATE BE
SET Saldo = 100
WHERE ID_BE = 2;
```

Modifica mezzo di pagamento usato da un *utente*

```
UPDATE Utente
SET Mezzo_di_pagamento = 'Satispay'
WHERE ID_U = 2;
```

Modifica sconto di un *piatto*

```
UPDATE Piatto
SET Sconto = 1.5
WHERE ID_P = 6;
```

Modifica nome di un *menu*

```
UPDATE Menu
SET Nome = 'Pizze tradizioni'
WHERE Nome = 'Pizze speciali';
```

Modifica valutazione di un *ristorante*

```
UPDATE Ristorante
SET Valutazione = 3.5
WHERE ID_R = 5;
```

Modifica appartenenza di un ristorante a una data *categoria*

```
UPDATE Appartiene
SET Categoria = 'Cucina internazionale'
WHERE ID_R = 2;
```

Modifica stato di un *rider*

```
UPDATE Rider
SET Stato = 'Disponibile'
WHERE Codice = 3;
```

Cancellazione

Cancellazione di un *utente*

```
DELETE FROM Utente  
WHERE ID_U = 5;
```

Cancellazione di un *rider*

```
DELETE FROM Rider  
WHERE Codice = 1;
```

Cancellazione di un *piatto*

```
DELETE FROM Piatto  
WHERE ID_P = 6;
```

Cancellazione di un *menu*

```
DELETE FROM Menu  
WHERE Nome = 'Menu dolci';
```

Cancellazione di un *ristorante*

```
DELETE FROM Ristorante  
WHERE ID_R = 3;
```

Cancellazione di un *ordine*

```
DELETE FROM Ordine  
WHERE ID_O = 3;
```

Cancellazione di una *categoria*

```
DELETE FROM Categoria  
WHERE Nome = 'Vegetariano';
```