***PROGETTO BASI DI DATI***

*di Giordano Davide*

Sommario

[1. Progettazione concettuale 2](#_Toc167824732)

[1.1 Requisiti Iniziali 2](#_Toc167824733)

[1.2 Requisiti Rivisti 4](#_Toc167824734)

[Frasi di carattere generale 4](#_Toc167824735)

[Frasi legate a Utente 4](#_Toc167824736)

[Frasi legate a Borsellino 5](#_Toc167824737)

[Frasi legate a Ristorante 5](#_Toc167824738)

[Frasi legate a Rider 6](#_Toc167824739)

[Frasi legate a Prodotto 6](#_Toc167824740)

[Frasi legate a Ordine 6](#_Toc167824741)

[1.3 Glossario dei Termini 7](#_Toc167824742)

[1.4 Business Rule e Modello ER 7](#_Toc167824743)

[2. Progettazione Logica 10](#_Toc167824744)

[2.1 Tavola dei volumi 10](#_Toc167824745)

[2.2 Tavola delle operazioni 11](#_Toc167824746)

[2.3 Ristrutturazione Schema ER 13](#_Toc167824747)

[2.3.1 Analisi Ridondanze 13](#_Toc167824748)

[2.3.2 Eliminazione Ridondanze 14](#_Toc167824749)

[2.3.3 Accorpamento entità e associazioni e generalizzazioni 18](#_Toc167824750)

[2.3.4 Eliminazione attributi composti/multi valore 18](#_Toc167824751)

[2.3.5 Scelta identificatori principali 19](#_Toc167824752)

[2.4 Schema ER ristrutturato + Business rule 19](#_Toc167824753)

[2.5 Schema Relazionale 22](#_Toc167824754)

[3 Implementazione SQL 25](#_Toc167824755)

# 1. Progettazione concettuale

## 1.1 Requisiti Iniziali

Immagine che contiene testo, Carattere, carta, schermata

Descrizione generata automaticamente

Immagine che contiene testo, schermata, menu, cibo

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente

## 1.2 Requisiti Rivisti

### Frasi di carattere generale

1. Si vuole 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 in bicicletta
2. Al momento dell’ordinazione il sistema cerca il rider libero con la somma minima della distanza dal ristorante più la distanza dell’utente
3. Una volta al mese vengono aggiornate le seguenti classifiche: Rider più veloci nel

consegnare gli ordini, prodotti più popolari, ristoranti con più recensioni positive,

utenti che hanno speso di più

### Frasi legate a Utente

1. Ogni utente deve registrarsi inserendo nome, e-mail, password numero di telefono, indirizzo di recapito
2. L’utente deve inserire un mezzo di pagamento e ricaricare il proprio borsellino

Elettronico

1. Gli utenti possono sottoscrivere la modalità premium che garantisce priorità sugli

ordini.

1. L’utente può collezionare codici sconto da utilizzare al momento dell’ordine sulla base

degli ordini effettuati in passato

1. Ogni utente può selezionare una lista di prodotti ed effettuare l’ordine
2. Nel profilo dell’utente si possono ispezionare gli ordini passati ed eventualmente

effettuare un reclamo

1. Dopo che l’ordine viene consegnato l’utente ha la possibilità di mandare messaggi sia

al ristorante che al rider

1. Dopo che l’ordine è stato consegnato l’utente può recensire sia il ristorante che il rider con una valutazione da 1 a 5 stelle e un commento testuale, è possibile dare una mancia al rider.

### Frasi legate a Borsellino

1. Il Borsellino ha un saldo che viene aggiornato ad ogni ordinazione

### Frasi legate a Ristorante

1. Ogni ristorante è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un’immagine del profilo e un numero di stellette aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell’ultima settimana
2. Ogni ristorante appartiene a una o più categorie in base al tipo di prodotto offerto
3. I ristoranti che dimostrano di saper garantire un ottimo servizio sono considerati Top Partner;
4. Per i ristoranti Top Partner di vuole tener traccia della data in cui sono entrati a far parte della categoria
5. I ristoranti propongono agli utenti una lista di prodotti da ordinare

### Frasi legate a Rider

1. Il sistema gestisce un numero arbitrario di riders
2. Ogni rider è identificato da un codice, dallo stato e dalla posizione aggiornata in tempo reale
3. I rider sono classificati in base al tipo di mezzo di trasporto
4. Per i rider che utilizzano monopattini elettrici è richiesto il numero di chilometri di autonomia del monopattino
5. Per monitorare le prestazioni dei rider si vuole tenere traccia del numero di consegne effettuato da ognuno, del momento in cui i prodotti da consegnare vengono affidati al rider e per gli ordini già consegnati, l’ora di recapito

### Frasi legate a Prodotto

1. Ogni prodotto ha un titolo, un’immagine, una lista di ingredienti, una lista di allergeni, il prezzo e un eventuale sconto
2. Ogni prodotto appartiene a una o più liste

### Frasi legate a Ordine

1. Gli ordini possono essere annullati finché non vengono affidati ad un rider

2. Gli ordini possono essere annullati sia dagli utenti che dai ristoranti

3. Per ordini che prevedono un tragitto totale superiore ai 10 chilometri solo i rider con la bici elettrica vengono interpellati

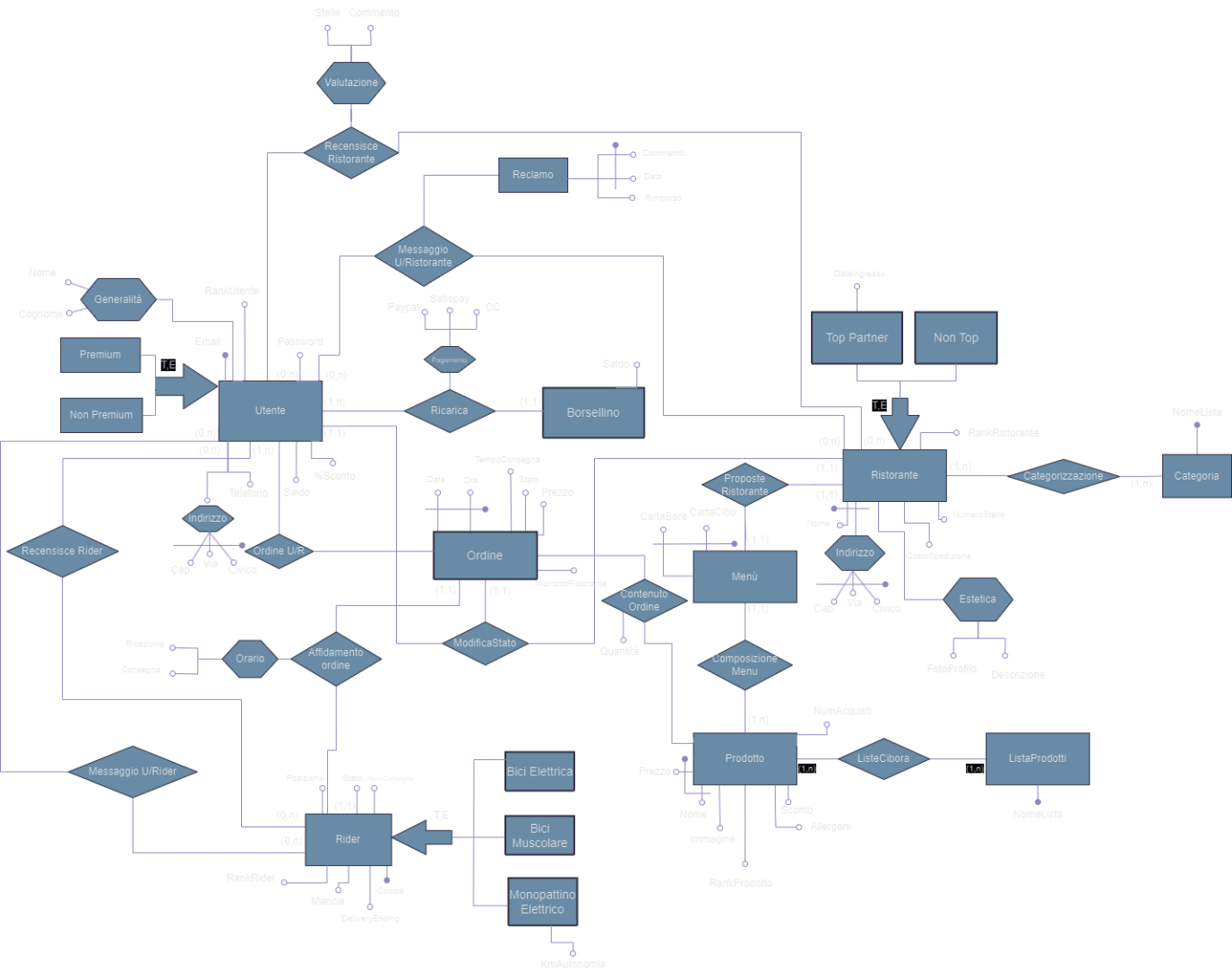
## 1.3 Glossario dei Termini

| **Concetto** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| --- | --- | --- | --- |
| **Utente** | Rappresenta un cliente che utilizza il servizio. | n/a | Borsellino, Reclamo, Ordine, Messaggio, Rider, Ristorante |
| **Rider** | Rappresenta il fattorino che consegna gli ordini ai clienti. | Ciclo Fattorino, fattorino | Utente, Ristorante, Ordine, Messaggio |
| **Ristoranti** | Rappresenta i ristoranti che offrono i loro servizi sulla piattaforma. | n/a | Utente, Categoria, Prodotto, Ordine, Rider, Reclamo |
| **Prodotto** | Rappresenta i prodotti offerti nei menu dei ristoranti. | Piatto, Portata, Cibo | Liste, Ristorante,  Ordine |
| **Borsellino** | Rappresenta il portafoglio virtuale dell'utente, utilizzato per pagare gli ordini. | Borsellino  Elettronico | Utente |
| **Menu** | Rappresenta il menu di un ristorante, che include le bevande e i cibi offerti. | n/a | Prodotto, Ristorante |
| **Reclamo** | Rappresenta un reclamo fatto dall'utente riguardo un ordine o un servizio. | n/a | Utente, Ristorante |
| **Categoria** | Rappresenta le diverse categorie di ristoranti o prodotti disponibili sulla piattaforma. | n/a | Ristorante |
| **Ordine** | Rappresenta un ordine effettuato da un utente. | n/a | Utente, Prodotto, Ristorante |

## 1.4 Business Rule e Modello ER

Si tenga conto delle seguenti Business Rule:

1. **Utenti e Ordini**
   * Un utente non può effettuare ordini fino alla prima ricarica del Borsellino.
   * Un ordine può essere effettuato solo se il suo costo totale è inferiore al saldo del borsellino dell'utente.
   * Gli utenti possono recensire ristoranti e rider solo dopo la consegna dell'ordine.
   * Gli utenti premium devono avere una priorità sugli ordini rispetto agli utenti non premium
   * Gli ordini possono essere annullati sia dai clienti sia dai ristoratori fino a quando non vengono affidati a un rider.
   * La sottoscrizione premium deve essere gestita con un sistema di rinnovo automatico o manuale.
2. **Borsellino Elettronico**
   * Il borsellino viene aggiornato a ogni transazione (es. pagamento di ordini, ricariche).
   * Il borsellino può essere ricaricato in ogni momento.
   * Il saldo minimo del borsellino necessario per effettuare un ordine deve essere maggiore o uguale al costo dell'ordine.
3. **Recensioni e Valutazioni**
   * Il valore di ogni attributo chiamato “Stelle” va da 1 a 5.
   * Le recensioni possono includere un commento testuale facoltativo.
4. **Percentuali e Sconti**
   * Il valore dell’attributo “%Sconto” di Utente prende valori percentuali da 0 a 40%.
   * I codici sconto accumulati dagli utenti devono essere gestiti in modo da applicare solo uno sconto per ordine.
5. **Top Partner**
   * Un ristorante può essere considerato “Top Partner” al raggiungimento dei seguenti obiettivi:
     + Almeno 20 ordini consegnati correttamente.
     + Valutazione media >= 4.5 stelle.
     + Percentuale di ordini annullati dal ristorante <= 1.5%.
     + Percentuale massima di ordini con reclami <= 2.5%.
6. **Gestione degli Ordini**
   * I rider vengono assegnati agli ordini in base alla somma minima della distanza "rider-ristorante" più "ristorante-cliente".
   * Se la distanza totale del tragitto (rider-ristorante-cliente) è >= 10 km, solo i rider con bici elettrica possono essere assegnati.
   * Gli utenti possono inviare reclami ai ristoranti tramite l'app.
7. **Gestione degli Ingredienti e Allergeni**
   * Ogni piatto deve includere una lista di ingredienti e allergeni.
   * Gli utenti possono filtrare i piatti in base agli allergeni.
8. **Gestione delle Mance**
   * Gli utenti possono aggiungere una mancia per i rider dopo la consegna dell'ordine.
   * Le mance devono essere registrate e visibili nel profilo del rider.
9. **Promozioni e Liste Speciali**
   * Le liste tematiche devono essere aggiornabili dai ristoranti.



# 2. Progettazione Logica

## 2.1 Tavola dei volumi

I dati si basano sulle stime di quelli di Glovo nel 2023, opportunamente riadattati considerando la natura prettamente alimentare della nostra applicazione (Glovo infatti consegna anche spesa, farmaceutica etc..). Dunque:

| **Concetto** | **Tipo** | **Volume** | **Descrizione** |
| --- | --- | --- | --- |
| Utente | E | 3,000,000 | Numero totale di utenti |
| Rider | E | 50,000 | Numero totale di rider registrati |
| Ristoranti | E | 100,000 | Numero totale di ristoranti registrati |
| Prodotto | E | 50,000 | Numero totale di prodotti registrati |
| Borsellino | E | 3,000,000 | Portafoglio dell’applicazione per l’utente |
| MetodoPagamento | E | 3,000,000 | Operazioni di aggiornamento saldo ad ogni transazione |
| Menù | E | 1,000 | Numero totale di menù registrati |
| Reclamo | E | 10,000,000 | Numero totale di reclami inseriti |
| Categoria | E | 200 | Numero totale di categorie registrate |
| Ordine | E | 5,000,000 | Numero totale di ordini inseriti |
| Affidamento ordine | R | 5,000,000 | Ordini affidati a rider ogni giorno |
| Modifica stato | R | 1,000,000 | Modifiche stato degli ordini ogni giorno |
| Ordine U/Ristorante | R | 5,000,000 | Relazioni ordine-ristorante inserite ogni giorno |
| Messaggio U/Ristorante | R | 50,000,000 | Messaggi scambiati tra utenti e ristoranti ogni giorno |
| Recensisce Ristorante | R | 150,000,000 | Recensioni ristoranti inserite ogni giorno |
| Ricarica | R | 200,000,000 | Operazioni di ricarica borsellino ogni giorno |
| Contenuto Ordine | R | 50,000 | Inserimento relazione piatto-menù ogni giorno |
| Composizione Menù | R | 50,000 | Inserimento relazione ingrediente-piatto ogni giorno |
| Liste “Cibora” | R | 50,000 | Inserimento nuovi elementi nelle liste tematiche di Cibora ogni giorno |
| Categorizzazione | R | 200 | Inserimento nuovi piatti nelle categorie ogni giorno |
| Recensisce Rider | R | 100,000,000 | Recensioni rider inserite ogni giorno |
| Proposte Ristorante | R | 1,000 | Nuove proposte piatti/menù da ristoranti ogni giorno |

## 2.2 Tavola delle operazioni

| **Numero dell'Operazione** | **Concetto** | **Frequenza** | **Modalità** |
| --- | --- | --- | --- |
| 1 | Nuovi utenti registrati | 8000 al giorno | I |
| 2 | Utenti cancellati | 1000 al giorno | I |
| 3 | Nuovi rider registrati | 50 al giorno | I |
| 4 | Rider cancellati | 10 al giorno | I |
| 5 | Nuovi ristoranti aggiunti | 100 al giorno | I |
| 6 | Ristoranti cancellati | 10 al giorno | I |
| 7 | Nuovi prodotti aggiunti | 500 al giorno | I |
| 8 | Prodotti cancellati | 50 al giorno | I |
| 9 | Aggiornamento saldo ad ogni transazione (ricarica o pagamento) | 300,000 al giorno | I |
| 10 | Nuovi menu aggiunti | 50 al giorno | I |
| 11 | Menu cancellati | 5 al giorno | I |
| 12 | Inserimento nuovo ordine | 500.000 al giorno | I |
| 13 | Aggiornamento Stato Ordine | 500.000 al giorno | B |
| 14 | Reclami inseriti | 50,000 al giorno | I |
| 15 | Reclami risolti e cancellati | 1,000 al giorno | I |
| 16 | Categorie aggiornate | 5 al giorno | B |
| 17 | Seleziona le categorie meno richieste e le cancella | 1 al mese | B |
| 18 | Aggiornamenti posizione ristoranti | 1 al mese | B |
| 18 | Aggiornamenti posizione utenti ogni settimana | 100,000 a settimana | B |
| 19 | Aggiornamenti posizione rider ogni settimana | 50,000 a settimana | B |
| 20 | Visualizzare ordini passati | 1,000,000 al giorno | I |
| 21 | Recensioni ristoranti inserite | 500,000 al giorno | I |
| 22 | Operazioni di ricarica borsellino | 200,000 a settimana | I |
| 23 | Inserimento relazione ingrediente-piatto | 5,000 al mese | I |
| 24 | Inserimento nuovi piatti nelle categorie | 50 a settimana | I |
| 25 | Recensioni rider inserite | 300,000 al giorno | I |
| 26 | Stampa dati dell’Utente | - | - |

## 2.3 Ristrutturazione Schema ER

### 2.3.1 Analisi Ridondanze

Analizziamo gli attributi delle entità e determiniamo se possono essere ricavati attraverso le associazioni. Consideriamo tutte le entità per vedere quali attributi possono essere ridondanti:

1. **Ordine**:
   * **Stato**: Considerata l’esistenza dell'associazione "ModificaStato", in cui vengono registrati tutti i cambiamenti di stato, i dati sono ricavabili dall’associazione;
   * **Prezzo**: Il prezzo dell'ordine può essere calcolato dalla somma dei prezzi dei prodotti inclusi nell'ordine (associazione "Contiene"), dunque questo attributo è ridondante
   * **Indirizzo Utente**: l’informazione è ricavabile dall’associazione di Ordine con Ristorante;
   * **TempoConsegna**: l’informazione è di fatto ottenibile attraverso la differenza di, attributi che troviamo nell’associazione tra Ordine e Rider, dunque è ridondante;
2. **Ristorante**:
   * **NumeroStelle**: Questo attributo è ridondante perché il numero di stelle può essere derivato dalle recensioni dei ristoranti (associazione "Recensione Ristorante").
3. **Prodotto**:
   * **NumAcquisti**: È ridondante poiché il numero di acquisti può essere calcolato sommando tutte le istanze dell'associazione con l'ordine.
4. **Rider**:
   * DeliveryEnding : l’informazione è di fatto ottenibile attraverso la differenza di, attributi che troviamo nell’associazione tra Ordine e Rider, dunque è ridondante;
5. **Borsellino**:
   * **Saldo**: Potrebbe essere ridondante in quanto il valore del saldo potrebbe essere calcolato come somma delle ricariche meno le spese;

### 2.3.2 Eliminazione Ridondanze

**Operazioni significative influenzate dalla ridondanza**

1. Inserimento di nuovi ordini
2. Aggiornamento dello stato dell'ordine (consegna completata)
3. Generazione di report sulle consegne

**Analisi delle operazioni**

**1. Inserimento di nuovi ordini**

**Schema di navigazione con ridondanza:**

1. Inserimento dell'ordine nella tabella "Ordine".
2. Calcolo del tempo di consegna e memorizzazione nelle tabelle "Ordine" e "Rider".

**Tavola degli accessi:**

| **Tipo** | **Entità** | **Accessi** | **Accessi Totali Giornalieri** |
| --- | --- | --- | --- |
| S | Ordine | 1 | 50,000 |
|  | Rider | 1 | 50,000 |
| Totale |  | 2 | 100,000 |

**Schema di navigazione senza ridondanza:**

1. Inserimento dell'ordine nella tabella "Ordine".

**Tavola degli accessi:**

| **Tipo** | **Entità** | **Accessi** | **Accessi Totali Giornalieri** |
| --- | --- | --- | --- |
| S | Ordine | 1 | 50,000 |
| Totale |  | 1 | 50,000 |

**Calcolo del numero di accessi:**

* **Con ridondanza:** 50,000 \* 2 \* 2 (Ordine e Rider) = 200,000 accessi al giorno
* **Senza ridondanza:** 50,000 \* 2 (Ordine) = 100,000 accessi al giorno

**2. Aggiornamento dello stato dell'ordine (consegna completata)**

**Schema di navigazione con ridondanza:**

1. Aggiornamento del tempo di consegna nelle tabelle "Ordine" e "Rider".

**Tavola degli accessi:**

| **Tipo** | **Entità** | **Accessi** | **Accessi Totali Giornalieri** |
| --- | --- | --- | --- |
| S | Ordine | 1 | 200,000 |
|  | Rider | 1 | 200,000 |
| Totale |  | 2 | 400,000 |

**Schema di navigazione senza ridondanza:**

1. Aggiornamento dello stato dell'ordine senza memorizzare il tempo di consegna.

**Tavola degli accessi:**

| **Tipo** | **Entità** | **Accessi** | **Accessi Totali Giornalieri** |
| --- | --- | --- | --- |
| S | Ordine | 1 | 200,000 |
| Totale |  | 1 | 200,000 |

**Calcolo del numero di accessi:**

* **Con ridondanza:** 200,000 \* 2 \* 2 (Ordine e Rider) = 800,000 accessi al giorno
* **Senza ridondanza:** 200,000 \* 2 (Ordine) = 400,000 accessi al giorno

**3. Generazione di report sulle consegne**

**Schema di navigazione con ridondanza:**

1. Recupero degli ordini dalla tabella "Ordine".
2. Recupero del tempo di consegna dalle tabelle "Ordine" e "Rider".

**Tavola degli accessi:**

| **Tipo** | **Entità** | **Accessi** | **Accessi Totali Giornalieri** |
| --- | --- | --- | --- |
| L | Ordine | 1 | 100,000 |
|  | Rider | 1 | 100,000 |
| Totale |  | 2 | 200,000 |

**Schema di navigazione senza ridondanza:**

1. Recupero degli ordini dalla tabella "Ordine".
2. Calcolo del tempo di consegna basato sulle ore di consegna e ricezione.

**Tavola degli accessi:**

| **Tipo** | **Entità** | **Accessi** | **Accessi Totali Giornalieri** |
| --- | --- | --- | --- |
| L | Ordine | 1 | 100,000 |
| Totale |  | 1 | 100,000 |

**Calcolo del numero di accessi:**

* **Con ridondanza:** 100,000 \* 2 (Ordine e Rider) = 200,000 accessi al giorno
* **Senza ridondanza:** 100,000 \* 1 (Ordine) = 100,000 accessi al giorno

**Spazio e tempo in presenza e assenza di ridondanza:**

**Calcolo dello spazio:**

* Supponiamo che l'attributo "TempoDiConsegna" occupi 4 byte (dato di tipo numerico).
* Numero di ordini giornalieri: 50,000
* Numero di ordini mensili: 50,000 \* 30 = 1,500,000
* Spazio aggiuntivo per ogni ordine: 4 byte
* Spazio aggiuntivo per ogni rider: 4 byte

**Spazio mensile con ridondanza:**

* 1,500,000 ordini \* 4 byte = 6,000,000 byte = 6 MB
* 1,500,000 rider \* 4 byte = 6,000,000 byte = 6 MB
* Spazio totale con ridondanza: 6 MB + 6 MB = 12 MB

**Calcolo degli accessi totali giornalieri:**

| **Operazione** | **Senza Ridondanza** | **Con Ridondanza** |
| --- | --- | --- |
| Inserimento di nuovi ordini | 50,000 | 100,000 |
| Aggiornamento dello stato dell'ordine | 200,000 | 400,000 |
| Generazione di report | 100,000 | 200,000 |
| Totale giornaliero | 350,000 | 700,000 |

**Conclusioni**

**Pro della ridondanza:**

* Riduzione del tempo di calcolo per le operazioni di visualizzazione e reportistica.

**Contro della ridondanza:**

* Aumento significativo del numero di accessi giornalieri (da 350,000 a 700,000).
* Aumento dello spazio di archiviazione (12 MB al mese).

Mantenere "TempoDiConsegna" solo nella tabella "Ordine" è efficiente sia in termini di spazio di archiviazione che di accessi giornalieri. Pertanto, l’attributo verrà eliminato dall’entità Rider.

### 2.3.3 Accorpamento entità e associazioni e generalizzazioni

Qui riportate ci sono le entità potenzialmente accorpabili e le generalizzazioni:

* **Utente e Premium/Non Premium:** Gli attributi di Premium e Non Premium possono essere rappresentati come un attributo booleano (vale la stessa cosa per Ristorante);
* **Recensione Ristorante e Valutazione:** La valutazione può essere parte dell'associazione di Recensione Ristorante.
* **Bici Elettrica, Bici Muscolare, Monopattino Elettrico:** Questi possono essere generalizzati in una super-entità "Veicolo" con attributi specifici per ciascun tipo di veicolo.

### 2.3.4 Eliminazione attributi composti/multi valore

Considerando questo schema ER e la definizione di attributi composti (=un attributo che può essere suddiviso in sotto-attributi) e multi-valore (=un attributo che può avere più valori per una singola entità), abbiamo come:

1. Attributi Composti:
   1. l’attributo “Indirizzo”, composto da "Via", "Città", "CAP", che compare sia nell’entità “Ristorante”, che in “Utente”;
   2. l’attributo “Generalità” in Utente, composto da Nome e Cognome;
   3. l’attributo “Valutazione” nella associazione “RecensisceRistorante”, composto da Stelle e Commento;
   4. l’attributo “Pagamento” in Ricarica, che contiene i diversi metodi di pagamento;
   5. l’attributo “Orario” in “Affidamento Ordine
   6. l’attributo “Estetica” in Ristorante, che rappresenta tutta la parte di presentazione di un ristorante sull’app;
2. Attributi Multi-valore:
   1. l’attributo Telefono in Utente e multi-valore, in quanto è possibile che uno stesso utente possa avere due numeri di telefono diversi;
   2. L’attributo “Foto Profilo” in Ristorante e multi-valore, in quanto è possibile mettere più di una foto del locale;
   3. l’attributo “Allergeni” in Prodotto e multi-valore, in quanto un prodotto può essere sia contenente arachidi che glutine;
   4. l’attributo “Commento” e multi-valore, in quanto una recensione potrebbe contenere più commenti (ad esempio, commenti aggiuntivi da parte dell'utente o risposte dal ristorante).

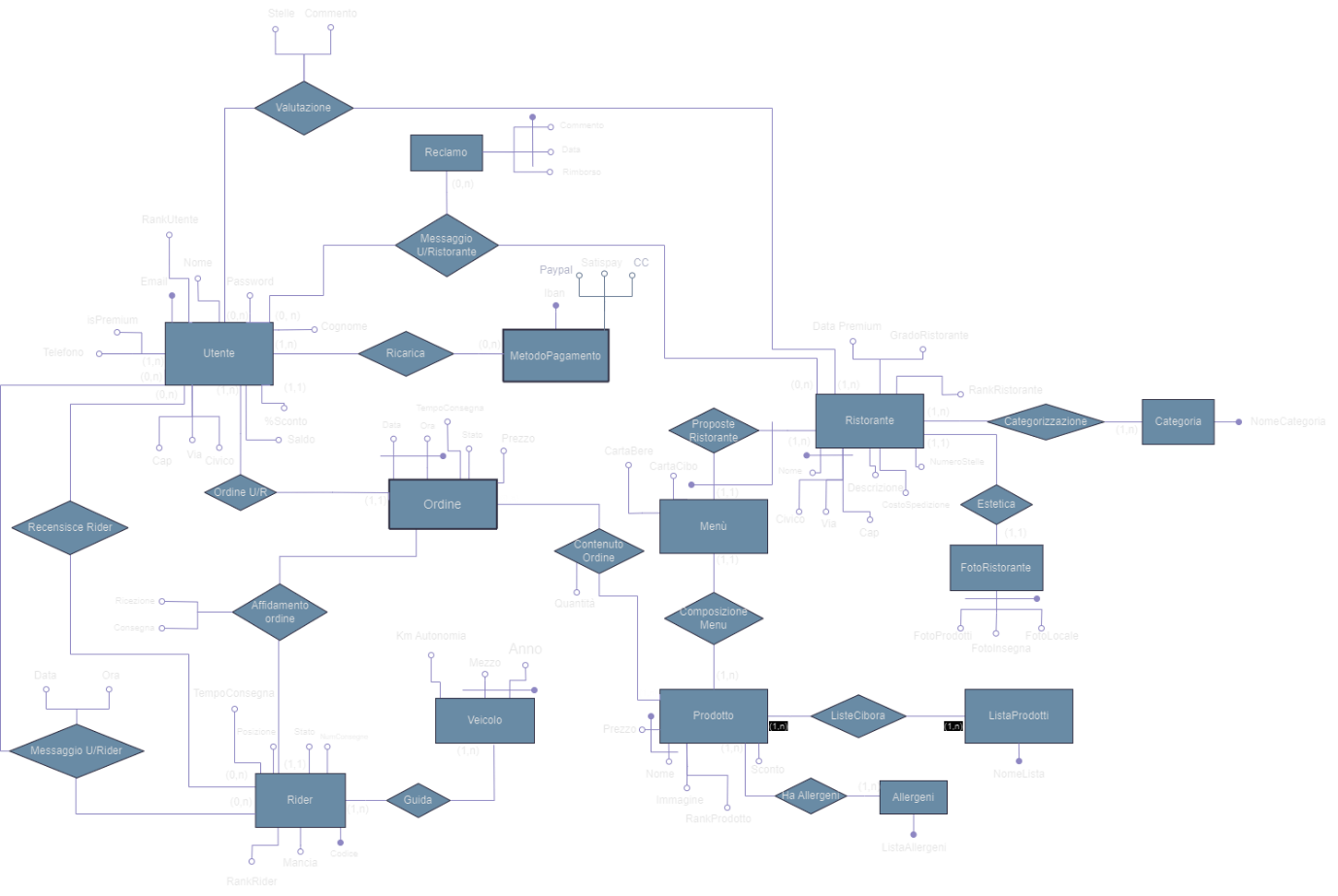
### 2.3.5 Scelta identificatori principali

Come identificatori principali abbiamo:

* **Utente**: E-mail
* **Ordine**: Data, Ora
* **Ristorante**: Nome, Cap, Via, Civico
* **Menu**: Carta Bere, Carta Cibo
* **Prodotto**: Nome, Prezzo
* **Lista Prodotti**: Nome Lista
* **Veicolo**: Mezzo, Anno
* **Rider**: Codice
* **Categoria**: Nome Categoria
* **Foto Ristorante**: Foto Prodotti, Foto Insegna, Foto Locale
* **Allergeni**: ListaAllergeni
* **Reclamo**: Data, Commento
* **MetodoPagamento**: IBAN

## 2.4 Schema ER ristrutturato + Business rule

Dunque, ecco come risulterà essere l’ER finale:



Si tenga conto delle seguenti Business Rule:

1. **Ristorante:**
   * Un ristorante può aggiornare il proprio menù solo una volta al giorno.
   * I ristoranti possono sospendere temporaneamente un piatto dal menu (ad esempio, se un ingrediente non è disponibile).
2. **Utenti e Ordini**
   * Un utente non può effettuare ordini fino alla prima ricarica del Borsellino.
   * Un ordine può essere effettuato solo se il suo costo totale è inferiore al saldo del borsellino dell'utente.
   * Gli utenti possono recensire ristoranti e rider solo dopo la consegna dell'ordine.
   * Gli utenti premium devono avere una priorità sugli ordini rispetto agli utenti non premium
   * Gli ordini possono essere annullati sia dai clienti sia dai ristoratori fino a quando non vengono affidati a un rider.
   * La sottoscrizione premium deve essere gestita con un sistema di rinnovo automatico o manuale.
3. **Autenticazione e Sicurezza**
   * Gli utenti devono confermare la loro identità tramite un'email di verifica al momento della registrazione.
   * Gli utenti devono essere notificati via email o SMS per ogni transazione effettuata (es. ricarica del borsellino, ordine effettuato, ordine annullato).
   * Le password degli utenti devono essere crittografate nel database
4. **Borsellino Elettronico**
   * Il borsellino viene aggiornato a ogni transazione (es. pagamento di ordini, ricariche).
   * Il borsellino può essere ricaricato in ogni momento.
   * Il saldo minimo del borsellino necessario per effettuare un ordine deve essere maggiore o uguale al costo dell'ordine.
5. **Recensioni e Valutazioni**
   * Il valore di ogni attributo chiamato “Stelle” va da 1 a 5.
   * Le recensioni possono includere un commento testuale facoltativo.
   * I reclami devono essere risolti entro un tempo limite specifico (ad esempio, 72 ore).
   * Gli utenti possono tracciare lo stato dei loro reclami attraverso l'app.
6. **Percentuali e Sconti**
   * Il valore dell’attributo “%Sconto” di Utente prende valori percentuali da 0 a 40%.
   * I codici sconto accumulati dagli utenti devono essere gestiti in modo da applicare solo uno sconto per ordine.
7. **Top Partner**
   * Un ristorante può essere considerato “Top Partner” al raggiungimento dei seguenti obiettivi:
     + Almeno 20 ordini consegnati correttamente.
     + Valutazione media >= 4.5 stelle.
     + Percentuale di ordini annullati dal ristorante <= 1.5%.
     + Percentuale massima di ordini con reclami <= 2.5%.
8. **Gestione degli Ordini**
   * I rider vengono assegnati agli ordini in base alla somma minima della distanza "rider-ristorante" più "ristorante-cliente".
   * Se la distanza totale del tragitto (rider-ristorante-cliente) è >= 10 km, solo i rider con bici elettrica possono essere assegnati.
   * Gli utenti possono inviare reclami ai ristoranti tramite l'app.
9. **Gestione degli Ingredienti e Allergeni**
   * Ogni piatto deve includere una lista di ingredienti e allergeni.
   * Gli utenti possono filtrare i piatti in base agli allergeni.
10. **Gestione delle Mance**
    * Gli utenti possono aggiungere una mancia per i rider dopo la consegna dell'ordine.
    * Le mance devono essere registrate e visibili nel profilo del rider.
11. **Rider**
    * I rider devono accettare o rifiutare un ordine entro un tempo limite specifico (ad esempio, 5 minuti) prima che l'ordine venga riassegnato.
    * I rider possono essere valutati dagli utenti solo una volta per ordine.
    * I rider devono confermare la consegna dell'ordine tramite l'applicazione per completare il processo di consegna.
12. **Promozioni e Liste Speciali**
    * Le liste tematiche devono essere aggiornabili dai ristoranti.

## 2.5 Schema Relazionale

Di seguito viene riportato lo schema relazionale:

1. **Utente(**E-mail, Nome, Cognome, Password, %Sconto, isPremium, Via, Civico, Cap, Saldo, Telefono, RankUtente**)**
2. **Ricarica(**Utente**,** Iban**)**

Utente(E-mail) referenzia Ricarica(Utente)

Metodo di Pagamento(Iban) referenzia Ricarica(Iban)

1. **Metodo Pagamento(**Iban, Satispay, CC, Paypal**)**
2. **Ordine(**Data, Ora, Utente, TempoConsegna, Stato, Prezzo, Rider**)**

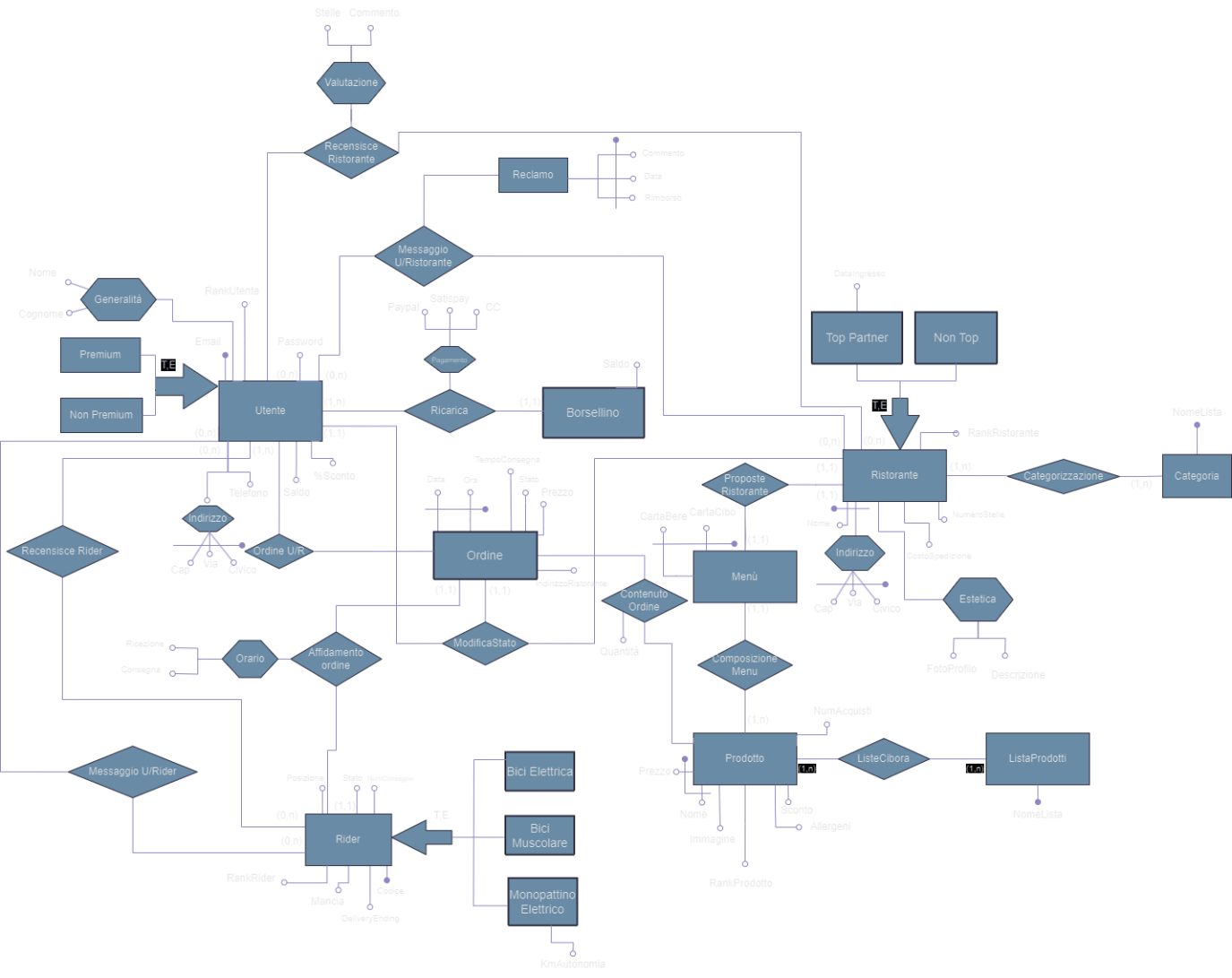
Rider(Codice) referenzia Ordine(Rider)

Utente(E-mail) referenzia Ordine(Utente)

1. **Ristorante(**Nome, Via, Civico, CAP, Descrizione, Costo Spedizione, NumeroStelle, Data Premium, Grado Ristorante, RankRistorante**)**
2. **Menu(**RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP, CartaCibo, CartaBere**)**

Ristorante(Nome, Via, Civico, CAP) referenzia Menu(RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP)

Prodotto(Prezzo, Nome) referenzia ContenutoOrdine(PrezzoP, NomeP)

1. **Prodotto(**Nome, Prezzo, Immagine, Sconto, RankProdotto**)**
2. **Categoria(**NomeCategoria**)**
3. **Categorizzazione(**RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP**,** NomeCategoria**)**

Ristorante(Nome, Via, Civico, CAP) referenzia Categoria(RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP)

Categoria(NomeCategoria) referenzia Categorizzazione(NomeCategoria)

1. **FotoRistorante(**FotoProdotti, FotoInsegna, FotoLocale, RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP**,)**

Ristorante**(**Nome, Via, Civico, CAP**)** referenziaFotoRistorante**(**RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP)

1. **Messaggio U/Ristorante(**Utente, RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP, Commento Reclamo, Data Reclamo**)**

Utente(E-mail) referenzia Messaggio U/Ristorante (Utente)

Ristorante(Nome, Via, Civico, CAP) referenzia Messaggio U/Ristorante(RistoranteN, Ristorate Via, RistoranteCivico, Ristorante CAP)

1. **Reclamo(**Commento, Data, Rimborso**)**
2. **Valutazione(**E-mail**,** RistoranteN, Ristorante Via, Ristorante Civico, Ristorante CAP**,** Stelle**,** Commento**)**

Utente(E-mail) referenzia Valutazione(E-mail)

Ristorante(Nome, Via, Civico, CAP) referenzia Valutazione(RistoranteN, Ristorate Via, RistoranteCivico, Ristorante CAP)

1. **Rider(**Codice, Posizione, Stato, NumConsegne, Mancia, RankRider**)**
2. **Guida(**Rider**,** Mezzo, Targa**)**

Rider(Codice) referenzia Guida(Rider)

Veicolo(Mezzo, Targa) referenzia Guida(Mezzo, Targa)

1. **Veicolo(**Mezzo, Anno, KmAutonomia**)**
2. **Messaggio U/Rider(**Utente, Rider, Data, Ora**)**

Utente(E-mail) referenzia MessaggioU/Rider(Utente)

Rider(Codice) referenzia MessaggioU/Rider(Rider)

1. **RecensiceRider(**Utente, Rider**)**

Utente(E-mail) referenzia MessaggioU/Rider(Utente)

Rider(Codice) referenzia MessaggioU/Rider(Rider)

1. **ListaCibora(**NomeLista, NomeP, PrezzoP**)**

ListaProdotti(ListaNome) referenzia ListaCibora(NomeLista)

Prodotto(Nome, Prezzo) referenzia ListaCibora(NomeP, PrezzoP)

1. **ListaProdotti(**ListaNome**)**
2. **Allergeni(**ListaAllergeni**)**
3. **HaAllergeni(**ListaAllergeni, PrezzoP, NomeP**)**

Prodotto(Prezzo, Nome) referenzia HaAllergeni(PrezzoP, NomeP)

Allergeni(ListaAllergeni) referenzia HaAllergeni(ListaAllergeni)

# 3 Implementazione SQL

Ecco i codici per creare il database (verificato su PGAdmin4):

-- Creazione delle tabelle

CREATE TABLE Utente (

Email VARCHAR(255) PRIMARY KEY,

Nome VARCHAR(255) NOT NULL,

Cognome VARCHAR(255) NOT NULL,

Password VARCHAR(255) NOT NULL,

PercentualeSconto INT DEFAULT 0,

isPremium BOOLEAN DEFAULT FALSE,

Via VARCHAR(255),

Civico VARCHAR(10),

Cap VARCHAR(10),

Saldo DECIMAL(10, 2) DEFAULT 0.00,

Telefono VARCHAR(15),

RankUtente INT DEFAULT 0

);

CREATE TABLE Rider (

Codice INT PRIMARY KEY,

Posizione VARCHAR(255),

Stato VARCHAR(50),

NumConsegne INT DEFAULT 0,

Mancia DECIMAL(10, 2) DEFAULT 0.00,

RankRider INT DEFAULT 0

);

CREATE TABLE Ordine (

Data DATE,

Ora TIME,

Utente VARCHAR(255),

TempoConsegna TIME,

Stato VARCHAR(50),

Prezzo DECIMAL(10, 2),

Rider INT,

PRIMARY KEY (Data, Ora, Utente),

FOREIGN KEY (Utente) REFERENCES Utente(Email),

FOREIGN KEY (Rider) REFERENCES Rider(Codice)

);

CREATE TABLE Ristorante (

Nome VARCHAR(255),

Via VARCHAR(255),

Civico VARCHAR(10),

CAP VARCHAR(10),

Descrizione TEXT,

CostoSpedizione DECIMAL(10, 2) DEFAULT 0.00,

NumeroStelle INT DEFAULT 0,

DataPremium DATE,

GradoRistorante INT DEFAULT 0,

RankRistorante INT DEFAULT 0,

PRIMARY KEY (Nome, Via, Civico, CAP)

);

CREATE TABLE Prodotto (

Nome VARCHAR(255),

Prezzo DECIMAL(10, 2) NOT NULL,

Immagine VARCHAR(255),

Sconto INT DEFAULT 0,

RankProdotto INT DEFAULT 0,

PRIMARY KEY (Nome, Prezzo)

);

CREATE TABLE MetodoPagamento (

Iban VARCHAR(34) PRIMARY KEY,

Satispay BOOLEAN DEFAULT FALSE,

CC BOOLEAN DEFAULT FALSE,

Paypal BOOLEAN DEFAULT FALSE

);

CREATE TABLE Ricarica (

Utente VARCHAR(255),

Iban VARCHAR(34),

PRIMARY KEY (Utente, Iban),

FOREIGN KEY (Utente) REFERENCES Utente(Email),

FOREIGN KEY (Iban) REFERENCES MetodoPagamento(Iban)

);

CREATE TABLE Menu (

RistoranteN VARCHAR(255),

RistoranteVia VARCHAR(255),

RistoranteCivico VARCHAR(10),

RistoranteCAP VARCHAR(10),

CartaCibo TEXT,

CartaBere TEXT,

PRIMARY KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP),

FOREIGN KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP) REFERENCES Ristorante(Nome, Via, Civico, CAP)

);

CREATE TABLE Categoria (

NomeCategoria VARCHAR(255) PRIMARY KEY

);

CREATE TABLE Categorizzazione (

RistoranteN VARCHAR(255),

RistoranteVia VARCHAR(255),

RistoranteCivico VARCHAR(10),

RistoranteCAP VARCHAR(10),

NomeCategoria VARCHAR(255),

PRIMARY KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP, NomeCategoria),

FOREIGN KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP) REFERENCES Ristorante(Nome, Via, Civico, CAP),

FOREIGN KEY (NomeCategoria) REFERENCES Categoria(NomeCategoria)

);

CREATE TABLE FotoRistorante (

FotoProdotti VARCHAR(255),

FotoInsegna VARCHAR(255),

FotoLocale VARCHAR(255),

RistoranteN VARCHAR(255),

RistoranteVia VARCHAR(255),

RistoranteCivico VARCHAR(10),

RistoranteCAP VARCHAR(10),

PRIMARY KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP),

FOREIGN KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP) REFERENCES Ristorante(Nome, Via, Civico, CAP)

);

CREATE TABLE MessaggioURistorante (

Utente VARCHAR(255),

RistoranteN VARCHAR(255),

RistoranteVia VARCHAR(255),

RistoranteCivico VARCHAR(10),

RistoranteCAP VARCHAR(10),

CommentoReclamo TEXT,

DataReclamo DATE,

PRIMARY KEY (Utente, RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP),

FOREIGN KEY (Utente) REFERENCES Utente(Email),

FOREIGN KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP) REFERENCES Ristorante(Nome, Via, Civico, CAP)

);

CREATE TABLE Reclamo (

Commento TEXT,

Data DATE,

Rimborso DECIMAL(10, 2)

);

CREATE TABLE Valutazione (

Email VARCHAR(255),

RistoranteN VARCHAR(255),

RistoranteVia VARCHAR(255),

RistoranteCivico VARCHAR(10),

RistoranteCAP VARCHAR(10),

Stelle INT,

Commento TEXT,

PRIMARY KEY (Email, RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP),

FOREIGN KEY (Email) REFERENCES Utente(Email),

FOREIGN KEY (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP) REFERENCES Ristorante(Nome, Via, Civico, CAP)

);

CREATE TABLE Veicolo (

Mezzo VARCHAR(50),

Anno INT,

KmAutonomia INT NULL,

PRIMARY KEY (Mezzo, Anno, Km)

);

CREATE TABLE Guida (

Rider INT,

Mezzo VARCHAR(50),

Targa VARCHAR(50),

PRIMARY KEY (Rider, Mezzo, Targa),

FOREIGN KEY (Rider) REFERENCES Rider(Codice),

FOREIGN KEY (Mezzo, Targa) REFERENCES Veicolo(Mezzo, Targa)

);

CREATE TABLE MessaggioURider (

Utente VARCHAR(255),

Rider INT,

Data DATE,

Ora TIME,

PRIMARY KEY (Utente, Rider, Data, Ora),

FOREIGN KEY (Utente) REFERENCES Utente(Email),

FOREIGN KEY (Rider) REFERENCES Rider(Codice)

);

CREATE TABLE RecensiceRider (

Utente VARCHAR(255),

Rider INT,

PRIMARY KEY (Utente, Rider),

FOREIGN KEY (Utente) REFERENCES Utente(Email),

FOREIGN KEY (Rider) REFERENCES Rider(Codice)

);

CREATE TABLE ListaProdotti (

NomeLista VARCHAR(255) PRIMARY KEY

);

CREATE TABLE ListaCibora (

NomeLista VARCHAR(255),

NomeP VARCHAR(255),

PrezzoP DECIMAL(10, 2),

PRIMARY KEY (NomeLista, NomeP, PrezzoP),

FOREIGN KEY (NomeLista) REFERENCES ListaProdotti(NomeLista),

FOREIGN KEY (NomeP, PrezzoP) REFERENCES Prodotto(Nome, Prezzo)

);

CREATE TABLE Allergeni (

ListaAllergeni VARCHAR(255) PRIMARY KEY

);

CREATE TABLE HaAllergeni (

ListaAllergeni VARCHAR(255),

PrezzoP DECIMAL(10, 2),

NomeP VARCHAR(255),

PRIMARY KEY (ListaAllergeni, PrezzoP, NomeP),

FOREIGN KEY (ListaAllergeni) REFERENCES Allergeni(ListaAllergeni),

FOREIGN KEY (PrezzoP, NomeP) REFERENCES Prodotto(Prezzo, Nome)

);

Ecco i comandi per il riempimento del database:

-- Inserimento dati nella tabella Prodotto

INSERT INTO Prodotto (Nome, Prezzo, Immagine, Sconto, RankProdotto)

VALUES

('Pizza Margherita', 8.50, 'pizza\_margherita.jpg', 10, 5),

('Sushi', 12.00, 'sushi.jpg', 5, 4),

('Insalata', 7.50, 'insalata.jpg', 0, 3),

('Gelato', 4.50, 'gelato.jpg', 0, 4);

-- Inserimento dati nella tabella Utente

INSERT INTO Utente (Email, Nome, Cognome, Password, PercentualeSconto, isPremium, Via, Civico, Cap, Saldo, Telefono, RankUtente)

VALUES

('utente1@example.com', 'Mario', 'Rossi', 'password1', 10, TRUE, 'Via Roma', '1', '00100', 100.00, '1234567890', 5),

('utente2@example.com', 'Luca', 'Bianchi', 'password2', 5, FALSE, 'Via Milano', '2', '20100', 50.00, '0987654321', 3);

-- Inserimento dati nella tabella Rider

INSERT INTO Rider (Codice, Posizione, Stato, NumConsegne, Mancia, RankRider)

VALUES

(1, 'Via Napoli', 'Disponibile', 100, 20.00, 4),

(2, 'Via Firenze', 'Non disponibile', 50, 10.00, 3);

-- Inserimento dati nella tabella Ordine

INSERT INTO Ordine (Data, Ora, Utente, TempoConsegna, Stato, Prezzo, Rider)

VALUES

('2023-05-27', '12:00:00', 'utente1@example.com', '00:30:00', 'Consegnato', 25.50, 1),

('2023-05-28', '13:00:00', 'utente2@example.com', '00:45:00', 'In preparazione', 40.00, 2);

-- Inserimento dati nella tabella Ristorante

INSERT INTO Ristorante (Nome, Via, Civico, CAP, Descrizione, CostoSpedizione, NumeroStelle, DataPremium, GradoRistorante, RankRistorante)

VALUES

('Ristorante A', 'Via Roma', '10', '00100', 'Cucina Italiana', 2.50, 5, '2023-01-01', 10, 8),

('Ristorante B', 'Via Milano', '20', '20100', 'Cucina Cinese', 3.00, 4, '2023-02-01', 8, 7);

-- Inserimento dati nella tabella MetodoPagamento

INSERT INTO MetodoPagamento (Iban, Satispay, CC, Paypal)

VALUES

('IT60X0542811101000000123456', TRUE, FALSE, FALSE),

('IT60X0542811101000000654321', FALSE, TRUE, TRUE);

-- Inserimento dati nella tabella Ricarica

INSERT INTO Ricarica (Utente, Iban)

VALUES

('utente1@example.com', 'IT60X0542811101000000123456'),

('utente2@example.com', 'IT60X0542811101000000654321');

-- Inserimento dati nella tabella Menu

INSERT INTO Menu (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP, CartaCibo, CartaBere)

VALUES

('Ristorante A', 'Via Roma', '10', '00100', 'Pizza, Pasta', 'Vino, Acqua'),

('Ristorante B', 'Via Milano', '20', '20100', 'Riso, Noodles', 'Tè, Birra');

-- Inserimento dati nella tabella Categoria

INSERT INTO Categoria (NomeCategoria)

VALUES

('Italiano'),

('Cinese');

-- Inserimento dati nella tabella Categorizzazione

INSERT INTO Categorizzazione (RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP, NomeCategoria)

VALUES

('Ristorante A', 'Via Roma', '10', '00100', 'Italiano'),

('Ristorante B', 'Via Milano', '20', '20100', 'Cinese');

-- Inserimento dati nella tabella FotoRistorante

INSERT INTO FotoRistorante (FotoProdotti, FotoInsegna, FotoLocale, RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP)

VALUES

('foto\_pizza.jpg', 'insegna\_ristorante\_a.jpg', 'locale\_ristorante\_a.jpg', 'Ristorante A', 'Via Roma', '10', '00100'),

('foto\_sushi.jpg', 'insegna\_ristorante\_b.jpg', 'locale\_ristorante\_b.jpg', 'Ristorante B', 'Via Milano', '20', '20100');

-- Inserimento dati nella tabella MessaggioURistorante

INSERT INTO MessaggioURistorante (Utente, RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP, CommentoReclamo, DataReclamo)

VALUES

('utente1@example.com', 'Ristorante A', 'Via Roma', '10', '00100', 'Ottimo servizio!', '2023-05-27'),

('utente2@example.com', 'Ristorante B', 'Via Milano', '20', '20100', 'Consegna in ritardo', '2023-05-28');

-- Inserimento dati nella tabella Reclamo

INSERT INTO Reclamo (Commento, Data, Rimborso)

VALUES

('Prodotto danneggiato', '2023-05-27', 5.00),

('Consegna in ritardo', '2023-05-28', 3.00);

-- Inserimento dati nella tabella Valutazione

INSERT INTO Valutazione (Email, RistoranteN, RistoranteVia, RistoranteCivico, RistoranteCAP, Stelle, Commento)

VALUES

('utente1@example.com', 'Ristorante A', 'Via Roma', '10', '00100', 5, 'Cibo delizioso!'),

('utente2@example.com', 'Ristorante B', 'Via Milano', '20', '20100', 3, 'Servizio medio');

-- Inserimento dati nella tabella Veicolo

INSERT INTO Veicolo (Mezzo, Anno, Km, Autonomia)

VALUES

('Bicicletta', 2021, NULL),

('Motorino', 2020, 150);

-- Inserimento dati nella tabella Guida

INSERT INTO Guida (Rider, Mezzo, Targa)

VALUES

(1, 'Bicicletta', 'ABC123'),

(2, 'Motorino', 'DEF456');

-- Inserimento dati nella tabella MessaggioURider

INSERT INTO MessaggioURider (Utente, Rider, Data, Ora)

VALUES

('utente1@example.com', 1, '2023-05-27', '12:00:00'),

('utente2@example.com', 2, '2023-05-28', '13:00:00');

-- Inserimento dati nella tabella RecensiceRider

INSERT INTO RecensiceRider (Utente, Rider)

VALUES

('utente1@example.com', 1),

('utente2@example.com', 2);

-- Inserimento dati nella tabella ListaProdotti

INSERT INTO ListaProdotti (NomeLista)

VALUES

('Lista Vegana'),

('Lista Dessert');

-- Inserimento dati nella tabella ListaCibora

INSERT INTO ListaCibora (NomeLista, NomeP, PrezzoP)

VALUES

('Lista Vegana', 'Insalata', 7.50),

('Lista Dessert', 'Gelato', 4.50);

-- Inserimento dati nella tabella Allergeni

INSERT INTO Allergeni (ListaAllergeni)

VALUES

('Glutine'),

('Lattosio');

-- Inserimento dati nella tabella HaAllergeni

INSERT INTO HaAllergeni (ListaAllergeni, PrezzoP, NomeP)

VALUES

('Glutine', 8.50, 'Pizza Margherita'),

('Lattosio', 4.50, 'Gelato');