

Progetto Laboratorio Basi di Dati

2023/2024

Relazione Piattaforma di food delivery

Partecipanti:

- Alessandro Demo | Turno T1 | Matricola 1049825

Indice:

1. Progettazione concettuale.....	1
1.1 Requisiti iniziali:.....	1
1.2 Glossario dei termini:.....	3
1.3 Requisiti rivisti:.....	6
1.4 Schema ER principale + Business Rules:.....	9
2. Progettazione logica.....	12
2.1 Tavola dei volumi:.....	12
2.2 Tavola delle operazioni:.....	15
2.3 Ristrutturazione dello schema ER:.....	17
2.3.1 Analisi ridondanza "Costo spedizione finale":.....	17
2.3.2 Eliminazione delle generalizzazioni:.....	21
2.3.3 Eventuale eliminazione degli attributi composti e degli attributi multivalore:.....	21
2.4 Schema ER ristrutturato + Business Rules:.....	22
2.5 Schema relazionale:.....	23
3. Implementazione.....	27
3.1 DDL di creazione del database:.....	27
3.2 DML di popolamento di tutte le tabelle del database:.....	36
3.3 Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne:.....	45
3.4 Qualche query per testare il corretto funzionamento del database:.....	47

1. Progettazione concettuale

1.1 Requisiti iniziali:

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

Per beneficiare del servizio, ogni utente deve registrarsi inserendo nome, email, password, numero di telefono, indirizzo di recapito. Una volta registrati, l'utente deve inserire un mezzo di pagamento (es.: carta di credito, paypal, satispay) e ricaricare il proprio borsellino elettronico. Il borsellino ha un saldo che viene aggiornato ad ogni ordinazione e l'utente può ricaricare il proprio borsellino in qualsiasi momento. Inoltre, gli utenti possono sottoscrivere la modalità premium che garantisce una priorità sugli ordini. L'utente può collezionare codici di sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.

Ogni ristorante (Figura 1(b)) è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un'immagine di profilo e un numero di stellette aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al tipo di cibo offerto (ad esempio: fast food, vegetariano, ...). I ristoranti che dimostrano di saper garantire un ottimo servizio (almeno 20 ordini consegnati correttamente, una valutazione clienti maggiore o uguale a 4.5 stelline su cinque, una percentuale massima di ordini annullati dal ristorante dell'1.5%, una percentuale massima di ordini con reclami del 2.5%) sono considerati Top Partner. I Top Partner compaiono in sezioni dedicate all'interno dell'app mobile Cibora e ricevono uno speciale badge che attesta il loro servizio eccellente, aiutando ad aumentare la credibilità e ottenere la fiducia dei clienti. Per i Top Partner si vuole tenere traccia della data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti una lista di piatti da ordinare. Ogni portata ha un titolo, un'immagine, una lista di ingredienti, una lista di allergeni, il

prezzo e un eventuale sconto. Inoltre, ogni piatto appartiene ad una o più liste (es. i più venduti, promozioni, dolci, salato, ecc.).

Ogni utente può selezionare una lista di 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.

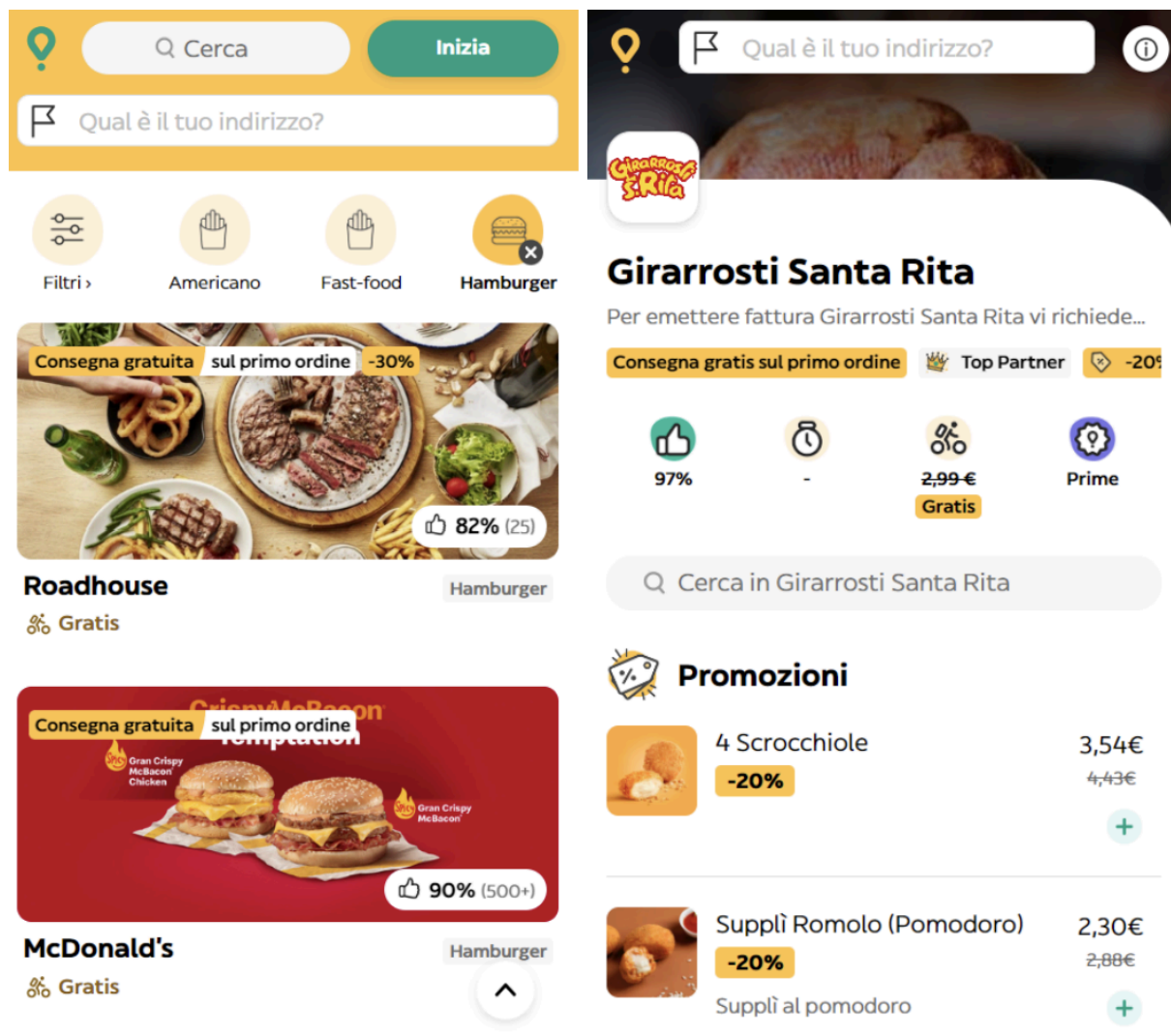


Figura 1 (a) La lista dei ristoranti con filtro “Hamburger”. (b) I dettagli di un ristorante.

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 (bici normale, bici 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	Descrizione	Sinonimi	Collegamenti
Ristorante	Cucina i piatti e prepara gli ordini	Ristoratori	Ordine, Piatto,

	da far consegnare al fattorino. Può essere un Top Partner.		Messaggio, Recensione, Reclamo, Classifica
Utente	Utente che effettua gli ordini dal ristorante. Può essere Premium o Base.	Cliente	Ordine, Borsellino elettronico, Sconto, Mezzo di pagamento, Messaggio, Reclamo, Classifica
Ordine	Ordine creato dal cliente, gestito dal ristorante e consegnato dal fattorino.		Ristorante, Utente, Fattorino, Piatto, Messaggio, Reclamo, Recensione
Fattorino	Fattorino che prende a carico gli ordini dal ristorante e li consegna all'utente.	Rider, Ciclofattorino	Ordine, Mezzo di trasporto, Messaggio, Recensione, Classifica
Borsellino elettronico	Borsellino presente all'interno dell'applicazione. Può essere caricato dall'utente.		Utente
Piatto	Piatti da inserire nell'ordine effettuato dall'utente. Il piatto viene creato dal ristorante.	Cibo, Portata, Pietanza	Ristorante, Ordine, Classifica

Sconto	Sconto da applicare al costo finale dell'ordine. Viene inserito dall'utente al momento dell'acquisto dell'ordine.	Codici di sconto	Utente
Mezzo di pagamento	Mezzo di pagamento da associare al borsellino da parte dell'utente. Può essere di diverso tipo (carta, Paypal, Satispay, ecc...).		Utente
Mezzo di trasporto	Mezzo di trasporto utilizzato dal fattorino per prendere l'ordine e consegnarlo all'utente. Può essere elettrico o normale.		Fattorino
Messaggio	Messaggio che l'utente può inviare al ristorante e/o al fattorino per ricevere informazioni sull'ordine.	Chat	Ristorante, Utente, Ordine, Fattorino
Reclamo	Reclamo che l'utente può effettuare nei confronti del ristorante da cui ha fatto un ordine precedentemente.		Ristorante, Utente, Ordine
Recensione	Recensione che può essere scritta dall'utente dopo che gli è stato consegnato l'ordine. La recensione		Ristorante, Ordine, Fattorino

	può essere fatta sia per il ristorante che per il fattorino, ai quali verranno assegnate delle stellette e un commento facoltativo.		
Classifica	Classifica a 10 posizioni da aggiornare mensilmente secondo dei certi criteri. La classifica si divide in 4 categorie: i migliori ristoranti, gli utenti che hanno speso di più, i fattorini più veloci e piatti più popolari.		Ristorante, Utente, Fattorino, Piatto

1.3 Requisiti rivisti:

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 in bicicletta.
L'utente deve iscriversi per utilizzare il servizio.

Per ogni **utente** rappresentiamo nome, email, password, numero di telefono e indirizzo di recapito. L'utente può scegliere se rimanere in modalità *Base* o passare a quella *Premium*, che garantisce una priorità sugli ordini. L'utente dal proprio profilo può gestire gli ordini, i reclami, gli sconti, il borsellino elettronico, il metodo di pagamento e i messaggi da inviare al fattorino e al ristorante. Una volta al mese viene aggiornata la classifica con gli utenti che hanno speso di più.

Il **borsellino elettronico** è disponibile per ogni utente ed è rappresentato da email utente e saldo, quest'ultimo viene aggiornato ad ogni ordinazione e può essere ricaricato in qualsiasi momento dall'utente.

Gli **sconti** vengono collezionati dall'utente in base al numero di ordini effettuati in passato e ne rappresentiamo valore economico, email utente, data di inizio validità, data di fine validità e codice.

Il **metodo di pagamento** viene inserito dall'utente e ne rappresentiamo email utente, mentre per le entità figlie rappresentiamo numero carta e CVV per la *carta*, email associata per *PayPal* e numero cellulare associato per *Satispay*.

Ogni **ristorante** è rappresentato da nome, indirizzo, descrizione, costo della spedizione, immagine di profilo e numero di stellette, quest'ultimo aggiornato ogni lunedì in base alla percentuale di recensioni positive dell'ultima settimana, inoltre viene anche rappresentato dall'appartenenza a una o più categorie in base al tipo di piatto offerto (ad esempio: *fast food*, *vegetariano*, ...).

I ristoranti che dimostrano di saper garantire un ottimo servizio, ossia almeno 20 ordini consegnati correttamente, una valutazione utenti maggiore o uguale a 4.5 stelline su cinque, una percentuale massima di ordini annullati dal ristorante dell'1.5%, una percentuale massima di ordini con reclami del 2.5%, sono considerati *Top Partner* e compaiono in sezioni dedicate dell'app mobile Cibora, ricevendo uno speciale badge che attesta il loro servizio eccellente, aiutando ad aumentare la credibilità e ottenere la fiducia degli utenti. Di questa tipologia di ristoranti si tiene traccia della data in cui sono entrati a far parte della categoria. Una volta al mese viene aggiornata la classifica dei ristoranti con più recensioni positive.

I ristoranti propongono agli utenti una lista di **piatti** da ordinare, di cui ne rappresentiamo titolo, nome ristorante, indirizzo ristorante, immagine, ingredienti, allergeni, lista di appartenenza (es. i più venduti, promozioni, dolci, salato, ecc.), prezzo ed eventuale sconto. Ogni mese viene fatta la classifica dei piatti più popolari.

L'**ordine** è rappresentato da email utente, nome ristorante, indirizzo ristorante, data creazione, orario creazione, id fattorino, indirizzo di consegna, prezzo finale, costo spedizione finale, orario presa a carico, orario consegna ed eventuale mancia. Al momento dell'ordine, il sistema trova il fattorino libero con la somma minima della distanza dal ristorante più la distanza dall'utente. Finché non sono affidati ad un fattorino per la consegna, gli ordini possono essere annullati sia dagli utenti, sia dai ristoratori, e per quelli che prevedano un tragitto "posizione corrente del fattorino-> ristorante-> utente" superiore ai 10 km, solo i fattorini con bici elettrica vengono interpellati.

Ogni **fattorino** è rappresentato da id fattorino, stato, posizione attuale e sono classificati in base al tipo di mezzo che utilizzano (bicicletta normale, bicicletta elettrica, monopattino). Se utilizzano un monopattino devono indicare l'autonomia. Si tiene traccia del numero di consegne effettuate, del momento in cui il piatto da consegnare viene affidato e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato all'utente. Ogni mese viene stilata la classifica dei fattorini più veloci nel consegnare ordini.

Il **mezzo di trasporto** è rappresentato da numero di serie e può essere *normale* o *elettrico*. Se elettrico ne rappresentiamo autonomia batteria.

Ogni **messaggio** è rappresentato da id_ordine, autore, data e ora. Dopo che l'ordine è stato effettuato l'utente ha la possibilità di inviarli sia al ristorante che al fattorino in caso ci fossero dei problemi con l'ordine, come mancata consegna o netto ritardo.

Il **reclamo** è rappresentato da id_ordine, testo, data e ora. Può essere creato dall'utente ispezionando gli ordini passati.

La **recensione** è rappresentata da id_ordine, data, ora, valutazione ed eventuale commento. Quando l'ordine è consegnato l'utente può recensire il ristorante e il fattorino.

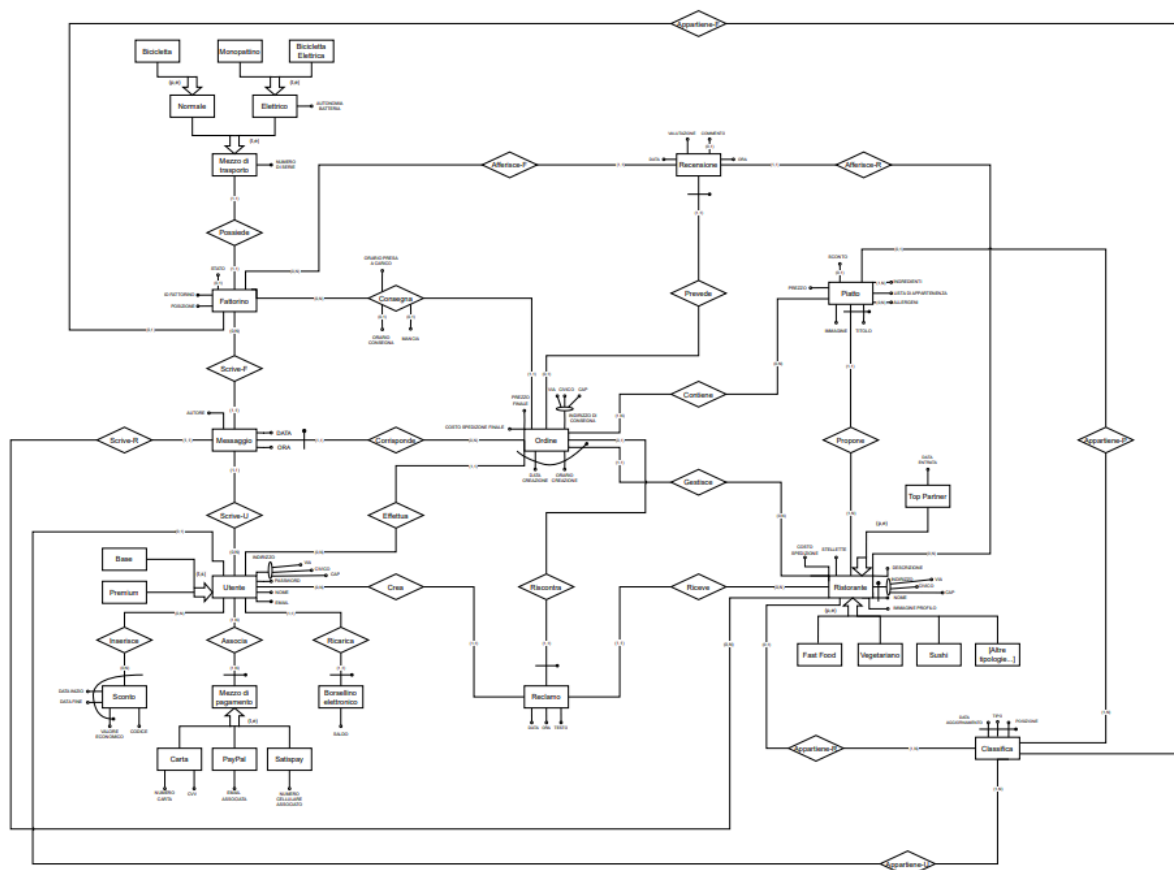
La **classifica** è rappresentata da data aggiornamento, posizione, tipo, id ristorante, id utente, id fattorino e id piatto. Essa viene aggiornata una volta al

mezzo secondo i criteri citati precedentemente per ristorante, utente, fattorino e piatto.

1.4 Schema ER principale + Business Rules:

Per lo schema ER principale vi si può accedere cliccando su questo link: ["Schema_ER_link"](#).

Per possibile necessità si allega anche la foto:



Ristorante:

Un ristorante diventa Top Partner se:

- ha gestito almeno 20 ordini che sono stati consegnati correttamente;
- ha una valutazione clienti maggiore o uguale a 4.5 stelline su 5;
- ha una percentuale massima di ordini annullati dal ristorante dell'1.5%;
- ha una percentuale massima di ordini con reclami del 2.5%.

Dopo essere diventato Top Partner e aver salvato la data di entrata, il ristorante compare in sezioni dedicate all'interno dell'app mobile Cibora e riceve uno speciale badge che attesta il suo servizio eccellente. Per generare elementi nell'entità Top Partner si utilizzano delle specifiche query.

L'attributo stellette viene aggiornato una volta a settimana ogni lunedì basandosi unicamente su recensioni positive, dunque con valutazioni composte da 3 stellette o più su 5.

Utente:

Un utente può sottoscrivere la modalità Premium pagando all'interno della applicazione per avere priorità sugli ordini.

Ordine:

L'ordine può essere annullato dall'utente o dal ristorante fino a che non viene preso in carico da un fattorino.

Quando si effettua l'ordine, il sistema trova il fattorino libero con la somma minima della distanza dal ristorante più la distanza dall'utente.

Per gli ordini che prevedano un tragitto "posizione corrente del fattorino -> ristorante -> utente" superiore ai 10 km, solo i fattorini con bici elettrica vengono interpellati.

Se ci sono problemi con l'ordine, l'utente può chattare, per chiarimenti su di esso, con ristorante e fattorino.

Fattorino:

Per tenere traccia del numero di consegne effettuate da un fattorino, si utilizza una query per analizzare il momento in cui il piatto da consegnare viene affidato ad un fattorino e l'ora in cui l'ordine è stato recapitato all'utente.

L'attributo stato di un fattorino può assumere uno di questi tre valori alla volta:

- 0, se il fattorino è occupato;
- 1, se il fattorino è disponibile;
- 2, se il fattorino è fuori servizio.

Piatto:

Un piatto fa parte di una determinata lista di appartenenza a discrezione del ristorante.

Il prezzo finale di un piatto scontato viene effettuato tramite l'operazione:
 $\text{prezzo del piatto} - (\text{prezzo del piatto} * \text{sconto} / 100)$.

Messaggio:

I messaggi vengono inviati dall'utente al ristorante o al fattorino per ricevere chiarimenti sull'ordine effettuato. Possono essere inviati dalla creazione dell'ordine fino alla confermata consegna di quest'ultimo.

Reclamo:

Il reclamo può essere inviato dall'utente nei confronti del ristorante solo dopo che è stato consegnato l'ordine all'utente stesso. L'analisi dei reclami viene effettuata per confermare la validità dell'entrata di un ristorante nella categoria dei Top Partner.

Recensione:

La recensione può essere scritta e inviata dall'utente solo dopo che gli è stato consegnato l'ordine effettuato. Con essa si assegnano le stellette utili a valutare il servizio del ristorante e del fattorino che hanno gestito e consegnato l'ordine.

Classifica:

La classifica viene aggiornata una volta al mese tramite query specifiche e vi sono presenti 4 categorie:

- fattorini più veloci nel consegnare gli ordini;
- piatti più popolari;
- ristoranti con più recensioni positive;
- utenti che hanno speso di più.

In queste 4 categorie sono presenti 10 posizioni occupate in base alle prestazioni mensili dei sopracitati.

2. Progettazione logica

2.1 Tavola dei volumi:

Concetto	Tipo	Volume	Motivazione
Ristorante	E	24.000	Statistiche condivise da Deliveroo
Utente	E	39.000.000	Statistiche condivise da Deliveroo
Ordine	E	Giornaliero: 1.200.000 Settimanale: 8.400.000 Mensile: 36.000.000 Annuale: 438.000.000	Stima tramite il calcolo: volume ristoranti * 50 ordini giornalieri di media per ristorante * numero di giorni
Fattorino	E	19.500	Statistiche condivise da Deliveroo
Borsellino elettronico	E	39.000.000	Ogni utente ha un proprio borsellino elettronico nell'applicazione
Piatto	E	1.080.000	Stima: ogni ristorante propone di media 45 piatti
Sconto	E	13.000.000	Stima: uno sconto per ogni tre utenti al mese
Mezzo di pagamento	E	39.000.000	Ogni utente registra almeno un metodo di pagamento
Mezzo di trasporto	E	19.500	Ogni fattorino ha almeno un mezzo di trasporto
Messaggio	E	4.400.000	Stima giornaliera: ogni utente invia un messaggio per un terzo degli

			ordini effettuati, ogni ristorante invia un messaggio e ogni fattorino invia almeno due messaggi
Reclamo	E	3.600.000	Stima annuale: un decimo degli ordini riscontra un reclamo
Recensione	E	12.000.000	Stima annuale: un terzo degli ordini prevede una recensione
Classifica	E	Mensile: 40 Annuale: 480	Stima: entrano a far parte 10 partecipanti al mese in ognuno dei 4 diversi tipi di categoria
Gestisce	A	Giornaliero: 1.200.000 Settimanale: 8.400.000 Mensile: 36.000.000 Annuale: 438.000.000	Stima: numero di ordini gestiti da tutti i ristoranti
Propone	A	1.080.000	Stima: ogni ristorante propone almeno 45 piatti diversi
Scrivo-R	A	1.200.000	Stima giornaliera: ogni ristorante invia almeno un messaggio ad ogni utente che ordina
Riceve	A	3.600.000	Stima annuale: i ristoranti ricevono i reclami per un decimo degli ordini effettuati
Afferisce-R	A	8.000.000	Stima annuale: ai ristoranti afferiscono due terzi delle recensioni totali
Effettua	A	Giornaliero: 1.200.000 Settimanale:	Stima: ordini effettuati da tutti gli utenti

		8.400.000 Mensile: 36.000.000 Annuale: 438.000.000	
Ricarica	A	13.000.000	Stima mensile: un terzo degli utenti ricarica una volta il proprio borsellino elettronico
Inserisce	A	19.500.000	Stima mensile: la metà degli utenti inserisce un codice sconto
Associa	A	39.000.000	Stima annuale: ogni utente associa un metodo di pagamento
Scrive-U	A	400.000	Stima giornaliera: ogni utente invia un messaggio per un terzo degli ordini effettuati
Crea	A	3.600.000	Stima annuale: gli utenti creano i reclami per un decimo degli ordini effettuati
Consegna	A	1.152.000	Stima giornaliera: tutti i fattorini consegnano il 96% degli ordini totali effettuati
Contiene	A	2.400.000	Stima giornaliera: ogni ordine contiene almeno due piatti
Corrisponde	A	4.800.000	Stima giornaliera: ad ogni ordine corrispondono tutti i messaggi inviati da utenti, ristoranti e fattorini per quello specifico ordine
Prevede	A	12.000.000	Stima annuale: un terzo degli ordini prevede una recensione
Riscontra	A	3.600.000	Stima annuale: un decimo degli ordini riscontra un reclamo

Possiede	A	19.500	Stima: ogni fattorino possiede almeno un mezzo di trasporto
Scrive-F	A	2.800.000	Stima giornaliera: ogni fattorino invia due messaggi per ogni ordine
Afferisce-F	A	4.000.000	Stima annuale: ai fattorini afferiscono un terzo delle recensioni totali
Appartiene-R	A	120	Stima annuale: ogni mese 10 ristoranti rientrano nella classifica
Appartiene-U	A	120	Stima annuale: ogni mese 10 utenti rientrano nella classifica
Appartiene-F	A	120	Stima annuale: ogni mese 10 fattorini rientrano nella classifica
Appartiene-P	A	120	Stima annuale: ogni mese 10 piatti rientrano nella classifica

2.2 Tavola delle operazioni:

Operazione	Descrizione	Tipo	Frequenza
1	Registrazione nuovo ristorante	I	125 al mese
2	Registrazione nuovo utente	I	250 al giorno
3	Registrazione nuovo fattorino	I	65 al mese
4	Registrazione nuovo piatto	I	5.625 al mese
5	Creazione nuovo ordine	I	1.200.000 al giorno
6	Inserimento sconto	I	650.000 al giorno

7	Ricarica borsellino	I	450.000 al giorno
8	Invio messaggio da ristorante	I	1.200.000 al giorno
9	Invio messaggio da utente	I	400.000 al giorno
10	Invio messaggio da fattorino	I	2.800.000 al giorno
11	Creazione reclamo	I	10.000 al giorno
12	Creazione recensione	I	400.000 al giorno
13	Presa a carico dell'ordine dal fattorino	B	1.152.000 al giorno
14	Aggiornamento ristoranti Top Partner	B	1 a settimana
15	Restituzione reclami totali per ogni ristorante	B	1 a settimana
16	Restituzione recensioni totali per ogni ristorante	B	1 a settimana
17	Restituzione recensioni totali per ogni fattorino	B	1 a settimana
18	Restituzione piatti totali offerti	B	2.400.000 al giorno
19	Restituzione dei fattorini liberi	B	615 al giorno *
20	Restituzione ordini totali effettuati	B	1.200.000 al giorno
21	Aggiornamento classifica	B	1 volta al mese
22	Determinazione costo di spedizione	B	1.200.000 al giorno

*1.200.000 ordini totali al giorno / 19.500 fattorini totali * 10 consegne effettuate per fattorino al giorno

2.3 Ristrutturazione dello schema ER:

2.3.1 Analisi ridondanza "Costo spedizione finale":

Tavola dei volumi:

Concetto	Tipo	Volume
Ristorante	E	24.000
Ordine	E	Giornaliero: 1.200.000 Settimanale: 8.400.000 Mensile: 36.000.000 Annuale: 438.000.000
Gestisce	A	Giornaliero: 1.200.000 Settimanale: 8.400.000 Mensile: 36.000.000 Annuale: 438.000.000

Tavola delle operazioni:

Operazione	Descrizione	Tipo	Frequenza
5	Creazione nuovo ordine	I	1.200.000 al giorno
20	Restituzione ordini totali effettuati	B	1.200.000 al giorno

22	Determinazione costo di spedizione	B	1.200.000 al giorno
----	------------------------------------	---	---------------------

- **Scenario A: Assenza ridondanza “Costo spedizione finale”:**

Schema di operazione:

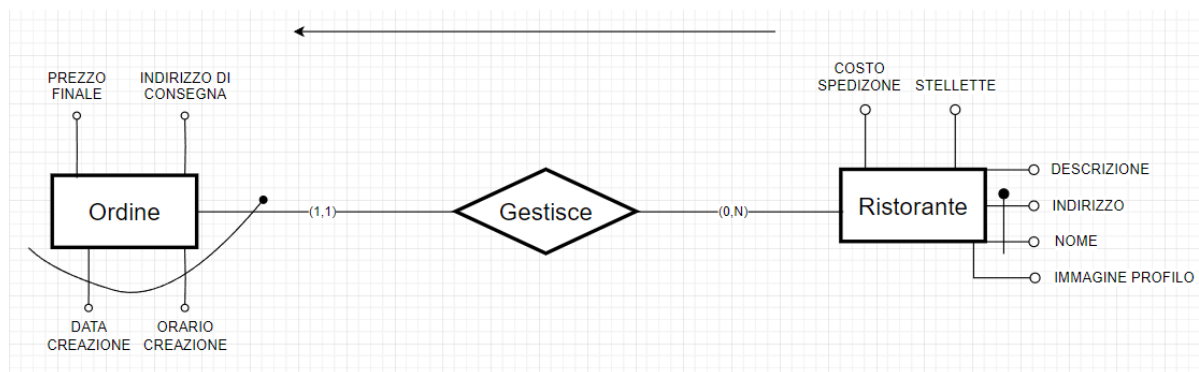


Tavola degli accessi:

Operazione	Concetto	Costrutto	Accessi	Tipo
5	Ristorante	Entità	0	/
5	Gestisce	Associazione	0	/
5	Ordine	Entità	1.200.000	S
20	Ristorante	Entità	1.200.000	L
20	Gestisce	Associazione	1.200.000	L
20	Ordine	Entità	1.200.000	S
22	Ristorante	Entità	1.200.000	S
22	Gestisce	Associazione	1.200.000	S
22	Ordine	Entità	1.200.000	L

Costi:

- **Spazio:** 0 byte
- **Tempo:**
 - Operazione 5: si effettua una scrittura per 1.200.000 ordini che vengono creati quotidianamente.
 - Operazione 20: si effettuano due letture per 1.200.000 ordini totali che i ristoranti ricevono e una scrittura per 1.200.000 ordini totali effettuati.
 - Operazione 22: si effettuano due scritture per 1.200.000 costi di spedizione determinati dal ristorante e una lettura per 1.200.000 ordini a cui vengono assegnati i costi di spedizione.
 - Totale: $1.200.000 + (2 * 1.200.000 + 1.200.000) + (2 * 1.200.000 + 1.200.000) = 8.400.000$ accessi al giorno.
- **Scenario B: Presenza ridondanza “Costo spedizione finale”:**

Schema di operazione:

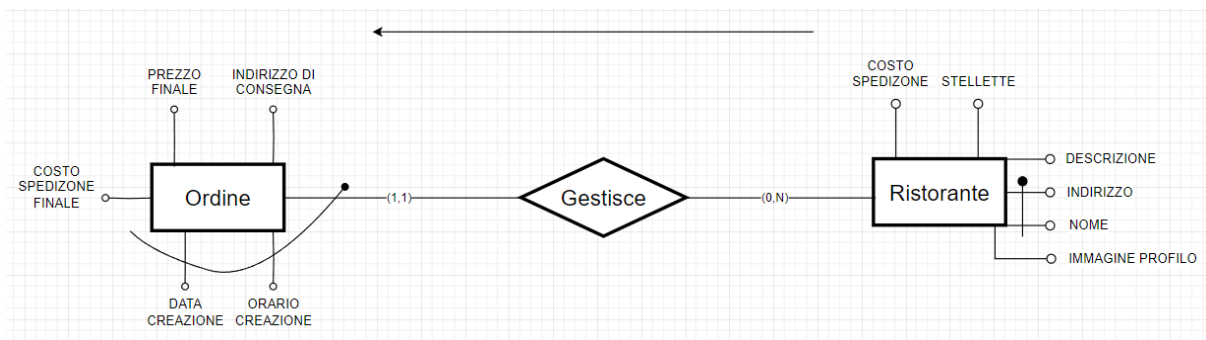


Tavola degli accessi:

Operazione	Concetto	Costrutto	Accessi	Tipo
5	Ristorante	Entità	0	/
5	Gestisce	Associazione	0	/
5	Ordine	Entità	1.200.000	S

20	Ristorante	Entità	1.200.000	L
20	Gestisce	Associazione	1.200.000	L
20	Ordine	Entità	1.200.000	S
22	Ristorante	Entità	1.200.000	S
22	Gestisce	Associazione	1.200.000	S
22	Ordine	Entità	1.200.000	L
22	Ordine	Entità	1.200.000	S

Costi:

- **Spazio:** si assume di utilizzare 4 byte per memorizzare il numero di costi di spedizione finale da determinare: $4 * 1.200.000 = 4.800.000$ byte al giorno = 4,8 Mbyte al giorno
144 Mbyte al mese
1753 Mbyte all'anno
- **Tempo:**
 - Operazione 5: si effettua una scrittura per 1.200.000 ordini che vengono creati quotidianamente.
 - Operazione 20: si effettuano due letture per 1.200.000 ordini totali che i ristoranti ricevono e una scrittura per 1.200.000 ordini totali effettuati.
 - Operazione 22: si effettuano tre scritture per 1.200.000 costi di spedizione determinati dal ristorante e dall'ordine, e una lettura per 1.200.000 ordini a cui vengono assegnati i costi di spedizione.
 - Totale: $1.200.000 + (2 * 1.200.000 + 1.200.000) + (3 * 1.200.000 + 1.200.000) = 9.600.000$ accessi al giorno.

Conclusioni e scelte:

In conclusione, si è scelto di rimuovere la ridondanza “costo spedizione finale” presente come attributo dell’entità Ordine perché la sua presenza richiederebbe di effettuare all’incirca 1.200.000 di accessi giornalieri in più, occupando anche più spazio in memoria, rispetto alla sua assenza.

2.3.2 Eliminazione delle generalizzazioni:

Cambiamento generalizzazioni: si è deciso di rimuovere le entità figlie di Utente, Ristorante, Mezzo di pagamento e Mezzo di trasporto nello schema ER ristrutturato perché non si differenziano in maniera particolare e contrastante dalle entità genitore, le quali sono rimaste con l’aggiunta degli attributi delle entità figlie, che potranno assumere anche valori NOT NULL.

2.3.3 Eventuale eliminazione degli attributi composti e degli attributi multivalore:

Attributi composti: l’attributo indirizzo presente per le entità Ristorante, Ordine e Utente è un composto dagli attributi via, civico e cap. Dunque, indirizzo viene scomposto portando alla presenza dei tre attributi appena citati nello schema ER ristrutturato.

Attributi multivalore: l’attributo ingredienti dell’entità Piatto può assumere 1 o più valori, dunque si è deciso di creare una nuova entità Ingredienti associata all’entità Piatto tramite Composizione. Inoltre, sono stati introdotti due nuovi identificatori nome e origine di arrivo per l’entità Ingredienti.

L’attributo allergeni dell’entità Piatto può assumere 0 o più valori, quindi si è deciso di formare una nuova entità Allergeni associata all’entità Piatto tramite Presenta. Sono stati introdotti anche un identificatore nome e un attributo descrizione per l’entità Allergeni.

Se un mezzo di trasporto è di tipo elettrico l'attributo autonomia è NOT NULL, altrimenti se il mezzo è di tipo normale l'attributo autonomia è NULL.

Mezzo di pagamento:

In base al tipo di mezzo di pagamento ci saranno degli attributi NOT NULL, che saranno:

- per la carta: numero carta e CVV;
 - per PayPal: email associata;
 - per Satispay: numero cellulare associato.
-

2.5 Schema relazionale:

Ristorante(Nome, Via, Civico, CAP, Descrizione, Stellette, Costo spedizione, Tipo di ristorante, Immagine profilo, Data entrata*, DataAggClassifica*, PosizClassifica*, TipoClassifica*)

- Ristorante(DataAggClassifica, PosizClassifica, TipoClassifica) referencia Classifica(Data aggiornamento, Posizione, Tipo)
- Vincolo unique su Ristorante(DataAggClassifica, PosizClassifica, TipoClassifica)

Utente(Email, Nome, Via, Civico, CAP, Password, Premium*, DataAggClassifica*, PosizClassifica*, TipoClassifica*)

- Utente(DataAggClassifica, PosizClassifica, TipoClassifica) referencia Classifica(Data aggiornamento, Posizione, Tipo)
- Vincolo unique su Utente(DataAggClassifica, PosizClassifica, TipoClassifica)

Ordine(Data creazione, Orario creazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist, Via, Civico, Cap, Prezzo finale, Fattorino*, Orario presa a carico*, Orario consegna*, Mancias*)

- Ordine(EmailUtente) referencia Utente(Email)
- Ordine(NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ristorante(Nome, Via, Civico, CAP)

- Ordine(Fattorino) referencia Fattorino(ID Fattorino)

Fattorino(ID Fattorino, Posizione, Stato*, Numero di serie, DataAggClassifica*, PosizClassifica*, TipoClassifica*)

- Fattorino(Numero di serie) referencia Mezzo di trasporto(Numero di serie)
- Fattorino(DataAggClassifica, PosizClassifica, TipoClassifica) referencia Classifica(Data aggiornamento, Posizione, Tipo)
- Vincolo unique su Fattorino(Numero di serie)
- Vincolo unique su Fattorino(DataAggClassifica, PosizClassifica, TipoClassifica)

Borsellino elettronico(Utente, Saldo)

- Borsellino elettronico(Utente) referencia Utente(Email)
- Vincolo unique su Borsellino(Utente)

Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist, Immagine, Prezzo, Lista di appartenenza, Sconto*, DataAggClassifica*, PosizClassifica*, TipoClassifica*)

- Piatto(NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ristorante(Nome, Via, Civico, CAP)
- Piatto(DataAggClassifica, PosizClassifica, TipoClassifica) referencia Classifica(Data aggiornamento, Posizione, Tipo)
- Vincolo unique su Piatto(DataAggClassifica, PosizClassifica, TipoClassifica)

Ingredienti(Nome, Origine di arrivo)

Allergeni(Nome, Descrizione)

Sconto(Utente, Valore economico, Data inizio, Data fine, Codice)

- Sconto(Utente) referencia Utente(Email)

Mezzo di pagamento(Utente, Tipo di pagamento, Numero carta*, CVV*, Email associata*, Numero cellulare associato*)

- Mezzo di pagamento(Utente) referencia Utente(Email)

Mezzo di trasporto(Numero di serie, Tipo di mezzo, Autonomia batteria*)

Messaggio(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist, Autore, Data, Ora, Fattorino)

- Messaggio(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ordine(Data creazione, Orario creazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)
- Messaggio(EmailUtente) referencia Utente(Email)
- Messaggio(NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ristorante(Nome, Via, Civico, CAP)
- Messaggio(Fattorino) referencia Fattorino(ID Fattorino)

Reclamo(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist, Data, Ora, Testo)

- Reclamo(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ordine(Data creazione, Orario creazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)
- Reclamo(EmailUtente) referencia Utente(Email)
- Reclamo(NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ristorante(Nome, Via, Civico, CAP)
- Vincolo unique su Reclamo(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)

Recensione(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist, Data, Ora, Valutazione, Commento*, Fattorino)

- Recensione(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ordine(Data creazione, Orario creazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)
- Recensione(EmailUtente) referencia Utente(Email)
- Recensione(NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ristorante(Nome, Via, Civico, CAP)
- Recensione(Fattorino) referencia Fattorino(ID Fattorino)
- Vincolo unique su Recensione(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)

Classifica(Data aggiornamento, Posizione, Tipo)

Associa(Utente, Tipo di pagamento)

- Associa(Utente, Tipo di pagamento) referencia Mezzo di pagamento(Utente, Tipo di pagamento)
- Associa(Utente) referencia Utente(Email)

Inserisce(Utente, Valore economico, Data inizio, Data fine)

- Inserisce(Utente, Valore economico, Data inizio, Data fine) referencia Sconto(Utente, Valore economico, Data inizio, Data fine)
- Inserisce(Utente) referencia Utente(Email)

Contiene(Titolo, DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)

- Contiene(DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Ordine(Data creazione, Orario creazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist)
- Contiene(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist)

Composizione(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist, Nome, Origine di arrivo)

- Composizione(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist)
- Composizione(Nome, Origine di arrivo) referencia Ingredienti(Nome, Origine di arrivo)

Presenta(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist, Nome)

- Presenta(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist) referencia Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist)
- Presenta(Nome) referencia Allergeni(Nome)

3. Implementazione

3.1 DDL di creazione del database:

Creazione della tabella Classifica

```
CREATE TABLE Classifica (  
    DataAggiornamento DATE,  
    Posizione INT,  
    Tipo VARCHAR(50),  
    PRIMARY KEY (DataAggiornamento, Posizione, Tipo)  
);
```

Creazione della tabella MezzoDiTrasporto

```
CREATE TABLE MezzoDiTrasporto (  
    NumeroDiSerie VARCHAR(50) PRIMARY KEY,  
    TipoDiMezzo VARCHAR(50),  
    AutonomiaBatteria DECIMAL(5, 2)  
);
```

Creazione della tabella Ristorante

```
CREATE TABLE Ristorante (  
    Nome VARCHAR(100),  
    Via VARCHAR(100),  
    Civico INT,  
    CAP CHAR(5),  
    Descrizione TEXT,  
    Stellette INT,  
    CostoSpedizione DECIMAL(5, 2),  
    TipoDiRistorante VARCHAR(50),  
    ImmagineProfilo TEXT,  
    DataEntrata DATE,
```

```

DataAggClassifica DATE,
PosizClassifica INT,
TipoClassifica VARCHAR(50),
PRIMARY KEY (Nome, Via, Civico, CAP),
FOREIGN KEY (DataAggClassifica, PosizClassifica, TipoClassifica)
REFERENCES Classifica(DataAggiornamento, Posizione, Tipo),
UNIQUE (DataAggClassifica, PosizClassifica, TipoClassifica)
);

```

Creazione della tabella Utente

```

CREATE TABLE Utente (
  Email VARCHAR(100) PRIMARY KEY,
  Nome VARCHAR(100),
  Via VARCHAR(100),
  Civico INT,
  CAP CHAR(5),
  Password VARCHAR(255),
  Premium BOOLEAN,
  DataAggClassifica DATE,
  PosizClassifica INT,
  TipoClassifica VARCHAR(50),
  FOREIGN KEY (DataAggClassifica, PosizClassifica, TipoClassifica)
REFERENCES Classifica(DataAggiornamento, Posizione, Tipo),
UNIQUE (DataAggClassifica, PosizClassifica, TipoClassifica)
);

```

Creazione della tabella Fattorino

```

CREATE TABLE Fattorino (
  IDFattorino INT PRIMARY KEY,
  Posizione VARCHAR(100),
  Stato VARCHAR(50),
  NumeroDiSerie VARCHAR(50) UNIQUE,

```

```

DataAggClassifica DATE,
PosizClassifica INT,
TipoClassifica VARCHAR(50),
FOREIGN KEY (NumeroDiSerie) REFERENCES
MezzoDiTrasporto(NumeroDiSerie),
FOREIGN KEY (DataAggClassifica, PosizClassifica, TipoClassifica)
REFERENCES Classifica(DataAggiornamento, Posizione, Tipo),
UNIQUE (DataAggClassifica, PosizClassifica, TipoClassifica)
);

```

Creazione della tabella Ordine

```

CREATE TABLE Ordine (
  DataCreazione DATE,
  OrarioCreazione TIME,
  EmailUtente VARCHAR(100),
  NomeRist VARCHAR(100),
  ViaRist VARCHAR(100),
  CivicoRist INT,
  CAPRist CHAR(5),
  Via VARCHAR(100),
  Civico INT,
  CAP CHAR(5),
  PrezzoFinale DECIMAL(10, 2),
  Fattorino INT,
  OrarioPresaCarico TIME,
  OrarioConsegna TIME,
  Mancia DECIMAL(5, 2),
  PRIMARY KEY (DataCreazione, OrarioCreazione, EmailUtente, NomeRist,
ViaRist, CivicoRist, CAPRist),
  FOREIGN KEY (EmailUtente) REFERENCES Utente(Email),
  FOREIGN KEY (NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES
Ristorante(Nome, Via, Civico, CAP),
  FOREIGN KEY (Fattorino) REFERENCES Fattorino(IDFattorino)

```

);

Creazione della tabella BorsellinoElettronico

```
CREATE TABLE BorsellinoElettronico (  
  Utente VARCHAR(100) PRIMARY KEY,  
  Saldo DECIMAL(10, 2),  
  FOREIGN KEY (Utente) REFERENCES Utente(Email),  
  UNIQUE (Utente)  
);
```

Creazione della tabella Piatto

```
CREATE TABLE Piatto (  
  Titolo VARCHAR(100),  
  NomeRist VARCHAR(100),  
  ViaRist VARCHAR(100),  
  CivicoRist INT,  
  CAPRist CHAR(5),  
  Immagine TEXT,  
  Prezzo DECIMAL(10, 2),  
  ListaDiAppartenenza VARCHAR(100),  
  Sconto DECIMAL(5, 2),  
  DataAggClassifica DATE,  
  PosizClassifica INT,  
  TipoClassifica VARCHAR(50),  
  PRIMARY KEY (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist),  
  FOREIGN KEY (NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES  
Ristorante(Nome, Via, Civico, CAP),  
  FOREIGN KEY (DataAggClassifica, PosizClassifica, TipoClassifica)  
REFERENCES Classifica(DataAggiornamento, Posizione, Tipo),  
  UNIQUE (DataAggClassifica, PosizClassifica, TipoClassifica)  
);
```

Creazione della tabella Ingredienti

```
CREATE TABLE Ingredienti (  
  Nome VARCHAR(100),  
  OrigineDiArrivo VARCHAR(100),  
  PRIMARY KEY (Nome, OrigineDiArrivo)  
);
```

Creazione della tabella Allergeni

```
CREATE TABLE Allergeni (  
  Nome VARCHAR(100) PRIMARY KEY,  
  Descrizione TEXT  
);
```

Creazione della tabella Sconto

```
CREATE TABLE Sconto (  
  Utente VARCHAR(100),  
  ValoreEconomico DECIMAL(5, 2),  
  DataInizio DATE,  
  DataFine DATE,  
  Codice VARCHAR(50),  
  PRIMARY KEY (Utente, ValoreEconomico, DataInizio, DataFine),  
  FOREIGN KEY (Utente) REFERENCES Utente(Email)  
);
```

Creazione della tabella MezzoDiPagamento

```
CREATE TABLE MezzoDiPagamento (  
  Utente VARCHAR(100),  
  TipoDiPagamento VARCHAR(50),  
  NumeroCarta VARCHAR(16),  
  CVV VARCHAR(4),
```



```
EmailAssociata VARCHAR(100),  
NumeroCellulareAssociato VARCHAR(15),  
PRIMARY KEY (Utente, TipoDiPagamento),  
FOREIGN KEY (Utente) REFERENCES Utente(Email)  
);
```

Creazione della tabella Messaggio

```
CREATE TABLE Messaggio (  
    DataOrdine DATE,  
    CreazioneOrdine TIME,  
    EmailUtente VARCHAR(100),  
    NomeRist VARCHAR(100),  
    ViaRist VARCHAR(100),  
    CivicoRist INT,  
    CAPRist CHAR(5),  
    Autore VARCHAR(100),  
    Data DATE,  
    Ora TIME,  
    Fattorino INT,  
    PRIMARY KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,  
ViaRist, CivicoRist, CAPRist),  
    FOREIGN KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,  
ViaRist, CivicoRist, CAPRist) REFERENCES Ordine(DataCreazione,  
OrarioCreazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist),  
    FOREIGN KEY (EmailUtente) REFERENCES Utente(Email),  
    FOREIGN KEY (NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES  
Ristorante(Nome, Via, Civico, CAP),  
    FOREIGN KEY (Fattorino) REFERENCES Fattorino(IDFattorino)  
);
```

Creazione della tabella Reclamo

```
CREATE TABLE Reclamo (  

```

```

DataOrdine DATE,
CreazioneOrdine TIME,
EmailUtente VARCHAR(100),
NomeRist VARCHAR(100),
ViaRist VARCHAR(100),
CivicoRist INT,
CAPRist CHAR(5),
Data DATE,
Ora TIME,
Testo TEXT,
PRIMARY KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,
ViaRist, CivicoRist, CAPRist),
FOREIGN KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,
ViaRist, CivicoRist, CAPRist) REFERENCES Ordine(DataCreazione,
OrarioCreazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist),
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email),
FOREIGN KEY (NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES
Ristorante(Nome, Via, Civico, CAP),
UNIQUE (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist,
CivicoRist, CAPRist)
);

```

Creazione della tabella Recensione

```

CREATE TABLE Recensione (
  DataOrdine DATE,
  CreazioneOrdine TIME,
  EmailUtente VARCHAR(100),
  NomeRist VARCHAR(100),
  ViaRist VARCHAR(100),
  CivicoRist INT,
  CAPRist CHAR(5),
  Data DATE,
  Ora TIME,

```

```

Valutazione INT,
Commento TEXT,
Fattorino INT,
PRIMARY KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,
ViaRist, CivicoRist, CAPRist),
FOREIGN KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,
ViaRist, CivicoRist, CAPRist) REFERENCES Ordine(DataCreazione,
OrarioCreazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist),
FOREIGN KEY (EmailUtente) REFERENCES Utente(Email),
FOREIGN KEY (NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES
Ristorante(Nome, Via, Civico, CAP),
FOREIGN KEY (Fattorino) REFERENCES Fattorino(IDFattorino),
UNIQUE (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist, ViaRist,
CivicoRist, CAPRist)
);

```

Creazione della tabella Associa

```

CREATE TABLE Associa (
  Utente VARCHAR(100),
  TipoDiPagamento VARCHAR(50),
  PRIMARY KEY (Utente, TipoDiPagamento),
  FOREIGN KEY (Utente, TipoDiPagamento) REFERENCES
MezzoDiPagamento(Utente, TipoDiPagamento),
  FOREIGN KEY (Utente) REFERENCES Utente(Email)
);

```

Creazione della tabella Inserisce

```

CREATE TABLE Inserisce (
  Utente VARCHAR(100),
  ValoreEconomico DECIMAL(5, 2),
  DataInizio DATE,
  DataFine DATE,

```

```
PRIMARY KEY (Utente, ValoreEconomico, DataInizio, DataFine),  
FOREIGN KEY (Utente) REFERENCES Utente(Email),  
FOREIGN KEY (Utente, ValoreEconomico, DataInizio, DataFine) REFERENCES  
Sconto(Utente, ValoreEconomico, DataInizio, DataFine)  
);
```

Creazione della tabella Contiene

```
CREATE TABLE Contiene (  
    Titolo VARCHAR(100),  
    DataOrdine DATE,  
    CreazioneOrdine TIME,  
    EmailUtente VARCHAR(100),  
    NomeRist VARCHAR(100),  
    ViaRist VARCHAR(100),  
    CivicoRist INT,  
    CAPRist CHAR(5),  
    PRIMARY KEY (Titolo, DataOrdine, CreazioneOrdine, EmailUtente,  
NomeRist, ViaRist, CivicoRist, CAPRist),  
    FOREIGN KEY (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,  
ViaRist, CivicoRist, CAPRist) REFERENCES Ordine(DataCreazione,  
OrarioCreazione, EmailUtente, NomeRist, ViaRist, CivicoRist, CAPRist),  
    FOREIGN KEY (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES  
Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist)  
);
```

Creazione della tabella Composizione

```
CREATE TABLE Composizione (  
    Titolo VARCHAR(100),  
    NomeRist VARCHAR(100),  
    ViaRist VARCHAR(100),  
    CivicoRist INT,  
    CAPRist CHAR(5),
```

```

NomeIngrediente VARCHAR(100),
OrigineDiArrivo VARCHAR(100),
PRIMARY KEY (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist,
NomeIngrediente, OrigineDiArrivo),
FOREIGN KEY (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES
Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist),
FOREIGN KEY (NomeIngrediente, OrigineDiArrivo) REFERENCES
Ingredienti(Nome, OrigineDiArrivo)
);

```

Creazione della tabella Presenta

```

CREATE TABLE Presenta (
Titolo VARCHAR(100),
NomeRist VARCHAR(100),
ViaRist VARCHAR(100),
CivicoRist INT,
CAPRist CHAR(5),
NomeAllergene VARCHAR(100),
PRIMARY KEY (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist,
NomeAllergene),
FOREIGN KEY (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist) REFERENCES
Piatto(Titolo, NomeRist, ViaRist, CivicoRist, CAPRist),
FOREIGN KEY (NomeAllergene) REFERENCES Allergeni(Nome)
);

```

3.2 DML di popolamento di tutte le tabelle del database:

Popolamento della tabella Classifica

```

INSERT INTO Classifica (DataAggiornamento, Posizione, Tipo) VALUES

```

```

('2024-06-01', 1, 'Ristorante'),
('2024-06-01', 2, 'Ristorante'),
('2024-06-01', 3, 'Ristorante'),
('2024-06-01', 4, 'Ristorante'),
('2024-06-01', 5, 'Ristorante'),
('2024-06-01', 1, 'Utente'),
('2024-06-01', 2, 'Utente'),
('2024-06-01', 3, 'Utente'),
('2024-06-01', 4, 'Utente'),
('2024-06-01', 5, 'Utente'),
('2024-06-01', 1, 'Piatto'),
('2024-06-01', 2, 'Piatto'),
('2024-06-01', 3, 'Piatto'),
('2024-06-01', 4, 'Piatto'),
('2024-06-01', 5, 'Piatto'),
('2024-06-01', 1, 'Fattorino'),
('2024-06-01', 2, 'Fattorino'),
('2024-06-01', 3, 'Fattorino'),
('2024-06-01', 4, 'Fattorino'),
('2024-06-01', 5, 'Fattorino');

```

Popolamento della tabella MezzoDiTrasporto

```

INSERT INTO MezzoDiTrasporto (NumeroDiSerie, TipoDiMezzo,
AutonomiaBatteria) VALUES
('AB123', 'Bicicletta', NULL),
('CD456', 'Bicicletta', NULL),
('EF789', 'Monopattino', 70.00),
('GH012', 'Bicicletta elettrica', 95.00),
('IJ345', 'Monopattino', 55.00);

```

Popolamento della tabella Ristorante

```

INSERT INTO Ristorante (Nome, Via, Civico, CAP, Descrizione, Stellette,
CostoSpedizione, TipoDiRistorante, ImmagineProfilo, DataEntrata,
DataAggClassifica, PosizClassifica, TipoClassifica) VALUES
('Ristorante A', 'Via Roma', 1, '00100', 'Ristorante italiano', 5, 2.50, 'Italiano',
'ristorante_a.jpg', '2023-01-01', '2024-06-01', 1, 'Ristorante'),
('Ristorante B', 'Via Milano', 2, '00200', 'Ristorante cinese', 4, 3.00, 'Cinese',
'ristorante_b.jpg', '2023-02-01', '2024-06-01', 2, 'Ristorante'),
('Ristorante C', 'Via Napoli', 3, '00300', 'Ristorante messicano', 3, 1.50,
'Messicano', 'ristorante_c.jpg', '2023-03-01', '2024-06-01', 3, 'Ristorante'),
('Ristorante D', 'Via Firenze', 4, '00400', 'Ristorante indiano', 4, 2.00, 'Indiano',
'ristorante_d.jpg', '2023-04-01', '2024-06-01', 4, 'Ristorante'),
('Ristorante E', 'Via Venezia', 5, '00500', 'Ristorante giapponese', 5, 3.50,
'Giapponese', 'ristorante_e.jpg', '2023-05-01', '2024-06-01', 5, 'Ristorante');

```

Popolamento della tabella Utente

```

INSERT INTO Utente (Email, Nome, Via, Civico, CAP, Password, Premium,
DataAggClassifica, PosizClassifica, TipoClassifica) VALUES
('utente1@example.com', 'Utente Uno', 'Via Alfa', 10, '00110', 'password1',
TRUE, '2024-06-01', 1, 'Utente'),
('utente2@example.com', 'Utente Due', 'Via Beta', 20, '00220', 'password2',
FALSE, '2024-06-01', 2, 'Utente'),
('utente3@example.com', 'Utente Tre', 'Via Gamma', 30, '00330',
'password3', TRUE, '2024-06-01', 3, 'Utente'),
('utente4@example.com', 'Utente Quattro', 'Via Delta', 40, '00440',
'password4', FALSE, '2024-06-01', 4, 'Utente'),
('utente5@example.com', 'Utente Cinque', 'Via Epsilon', 50, '00550',
'password5', TRUE, '2024-06-01', 5, 'Utente');

```

Popolamento della tabella Fattorino

```
INSERT INTO Fattorino (IDFattorino, Posizione, Stato, NumeroDiSerie,
DataAggClassifica, PosizClassifica, TipoClassifica) VALUES
(1, 'Zona A', 'Disponibile', 'AB123', '2024-06-01', 1, 'Fattorino'),
(2, 'Zona B', 'Disponibile', 'CD456', '2024-06-01', 2, 'Fattorino'),
(3, 'Zona C', 'Non Disponibile', 'EF789', '2024-06-01', 3, 'Fattorino'),
(4, 'Zona D', 'Disponibile', 'GH012', '2024-06-01', 4, 'Fattorino'),
(5, 'Zona E', 'Non Disponibile', 'IJ345', '2024-06-01', 5, 'Fattorino');
```

Popolamento della tabella Ordine

```
INSERT INTO Ordine (DataCreazione, OrarioCreazione, EmailUtente,
NomeRist, ViaRist, CivicoRist, CAPRist, Via, Civico, CAP, PrezzoFinale,
Fattorino, OrarioPresaCarico, OrarioConsegna, Mancias) VALUES
('2024-06-01', '12:00:00', 'utente1@example.com', 'Ristorante A', 'Via Roma',
1, '00100', 'Via Alfa', 10, '00110', 25.50, 1, '12:10:00', '12:30:00', 2.00),
('2024-06-01', '13:00:00', 'utente2@example.com', 'Ristorante B', 'Via
Milano', 2, '00200', 'Via Beta', 20, '00220', 30.00, 2, '13:10:00', '13:40:00',
3.00),
('2024-06-01', '14:00:00', 'utente3@example.com', 'Ristorante C', 'Via Napoli',
3, '00300', 'Via Gamma', 30, '00330', 20.00, 3, '14:10:00', '14:35:00', 1.50),
('2024-06-01', '15:00:00', 'utente4@example.com', 'Ristorante D', 'Via
Firenze', 4, '00400', 'Via Delta', 40, '00440', 35.00, 4, '15:10:00', '15:50:00',
4.00),
('2024-06-01', '16:00:00', 'utente5@example.com', 'Ristorante E', 'Via
Venezia', 5, '00500', 'Via Epsilon', 50, '00550', 40.00, 5, '16:10:00', '16:45:00',
5.00);
```

Popolamento della tabella BorsellinoElettronico

```
INSERT INTO BorsellinoElettronico (Utente, Saldo) VALUES
('utente1@example.com', 50.00),
('utente2@example.com', 75.00),
('utente3@example.com', 100.00),
('utente4@example.com', 125.00),
```



```
('utente5@example.com', 150.00);
```

Popolamento della tabella Piatto

```
INSERT INTO Piatto (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist, Immagine,
Prezzo, ListaDiAppartenenza, Sconto, DataAggClassifica, PosizClassifica,
TipoClassifica) VALUES
('Pizza Margherita', 'Ristorante A', 'Via Roma', 1, '00100',
'pizza_margherita.jpg', 8.50, 'Menu Principale', 0.00, '2024-06-01', 1, 'Piatto'),
('Sushi', 'Ristorante B', 'Via Milano', 2, '00200', 'sushi.jpg', 12.00, 'Menu
Principale', 0.00, '2024-06-01', 2, 'Piatto'),
('Taco', 'Ristorante C', 'Via Napoli', 3, '00300', 'taco.jpg', 5.00, 'Menu
Principale', 0.00, '2024-06-01', 3, 'Piatto'),
('Curry', 'Ristorante D', 'Via Firenze', 4, '00400', 'curry.jpg', 10.00, 'Menu
Principale', 0.00, '2024-06-01', 4, 'Piatto'),
('Ramen', 'Ristorante E', 'Via Venezia', 5, '00500', 'ramen.jpg', 15.00, 'Menu
Principale', 0.00, '2024-06-01', 5, 'Piatto');
```

Popolamento della tabella Ingredienti

```
INSERT INTO Ingredienti (Nome, OrigineDiArrivo) VALUES
('Pomodoro', 'Italia'),
('Mozzarella', 'Italia'),
('Riso', 'Giappone'),
('Avocado', 'Messico'),
('Pollo', 'India');
```

Popolamento della tabella Allergeni

```
INSERT INTO Allergeni (Nome, Descrizione) VALUES
('Glutine', 'Presente nei cereali'),
('Lattosio', 'Presente nei latticini'),
('Pesce', 'Presente nei prodotti ittici'),
```

```
('Uova', 'Presente nelle uova'),  
( 'Arachidi', 'Presente nelle arachidi');
```

Popolamento della tabella Sconto

```
INSERT INTO Sconto (Utente, ValoreEconomico, DataInizio, DataFine, Codice)  
VALUES  
( 'utente1@example.com', 5.00, '2024-06-01', '2024-06-30', 'SCONTO5'),  
( 'utente2@example.com', 10.00, '2024-06-01', '2024-06-30', 'SCONTO10'),  
( 'utente3@example.com', 15.00, '2024-06-01', '2024-06-30', 'SCONTO15'),  
( 'utente4@example.com', 20.00, '2024-06-01', '2024-06-30', 'SCONTO20'),  
( 'utente5@example.com', 25.00, '2024-06-01', '2024-06-30', 'SCONTO25');
```

Popolamento della tabella MezzoDiPagamento

```
INSERT INTO MezzoDiPagamento (Utente, TipoDiPagamento, NumeroCarta,  
CVV, EmailAssociata, NumeroCellulareAssociato) VALUES  
( 'utente1@example.com', 'Carta di Credito', '1234567890123456', '123',  
NULL, NULL),  
( 'utente2@example.com', 'PayPal', NULL, NULL, 'utente2@example.com',  
NULL),  
( 'utente3@example.com', 'Satispay', NULL, NULL, NULL, '3456789012'),  
( 'utente4@example.com', 'Carta di Credito', '4567890123456789', '456',  
NULL, NULL),  
( 'utente5@example.com', 'Carta di Credito', '5678901234567890', '567',  
NULL, NULL);
```

Popolamento della tabella Messaggio

```
INSERT INTO Messaggio (DataOrdine, CreazioneOrdine, EmailUtente,  
NomeRist, ViaRist, CivicoRist, CAPRist, Autore, Data, Ora, Fattorino) VALUES  
( '2024-06-01', '12:00:00', 'utente1@example.com', 'Ristorante A', 'Via Roma',  
1, '00100', 'Utente Uno', '2024-06-01', '12:15:00', 1),
```

```
( '2024-06-01', '13:00:00', 'utente2@example.com', 'Ristorante B', 'Via
Milano', 2, '00200', 'Utente Due', '2024-06-01', '13:20:00', 2),
( '2024-06-01', '14:00:00', 'utente3@example.com', 'Ristorante C', 'Via Napoli',
3, '00300', 'Utente Tre', '2024-06-01', '14:25:00', 3),
( '2024-06-01', '15:00:00', 'utente4@example.com', 'Ristorante D', 'Via
Firenze', 4, '00400', 'Utente Quattro', '2024-06-01', '15:30:00', 4),
( '2024-06-01', '16:00:00', 'utente5@example.com', 'Ristorante E', 'Via
Venezia', 5, '00500', 'Utente Cinque', '2024-06-01', '16:35:00', 5);
```

Popolamento della tabella Reclamo

```
INSERT INTO Reclamo (DataOrdine, CreazioneOrdine, EmailUtente, NomeRist,
ViaRist, CivicoRist, CAPRist, Data, Ora, Testo) VALUES
( '2024-06-01', '12:00:00', 'utente1@example.com', 'Ristorante A', 'Via Roma',
1, '00100', '2024-06-01', '12:45:00', 'Ordine in ritardo'),
( '2024-06-01', '13:00:00', 'utente2@example.com', 'Ristorante B', 'Via
Milano', 2, '00200', '2024-06-01', '13:50:00', 'Ordine sbagliato'),
( '2024-06-01', '14:00:00', 'utente3@example.com', 'Ristorante C', 'Via Napoli',
3, '00300', '2024-06-01', '14:55:00', 'Mancava un piatto'),
( '2024-06-01', '15:00:00', 'utente4@example.com', 'Ristorante D', 'Via
Firenze', 4, '00400', '2024-06-01', '15:55:00', 'Cibo freddo'),
( '2024-06-01', '16:00:00', 'utente5@example.com', 'Ristorante E', 'Via
Venezia', 5, '00500', '2024-06-01', '16:50:00', 'Qualità scadente');
```

Popolamento della tabella Recensione

```
INSERT INTO Recensione (DataOrdine, CreazioneOrdine, EmailUtente,
NomeRist, ViaRist, CivicoRist, CAPRist, Data, Ora, Valutazione, Commento,
Fattorino) VALUES
( '2024-06-01', '12:00:00', 'utente1@example.com', 'Ristorante A', 'Via Roma',
1, '00100', '2024-06-01', '12:50:00', 5, 'Ottimo servizio', 1),
( '2024-06-01', '13:00:00', 'utente2@example.com', 'Ristorante B', 'Via
Milano', 2, '00200', '2024-06-01', '13:55:00', 4, 'Buon cibo', 2),
```

```
('2024-06-01', '14:00:00', 'utente3@example.com', 'Ristorante C', 'Via Napoli',
3, '00300', '2024-06-01', '15:00:00', 3, 'Servizio lento', 3),
('2024-06-01', '15:00:00', 'utente4@example.com', 'Ristorante D', 'Via
Firenze', 4, '00400', '2024-06-01', '16:00:00', 2, 'Cibo freddo', 4),
('2024-06-01', '16:00:00', 'utente5@example.com', 'Ristorante E', 'Via
Venezia', 5, '00500', '2024-06-01', '17:00:00', 1, 'Non soddisfatto', 5);
```

Popolamento della tabella Associa

```
INSERT INTO Associa (Utente, TipoDiPagamento) VALUES
('utente1@example.com', 'Carta di Credito'),
('utente2@example.com', 'PayPal'),
('utente3@example.com', 'Satispay'),
('utente4@example.com', 'Carta di Credito'),
('utente5@example.com', 'Carta di Credito');
```

Popolamento della tabella Inserisce

```
INSERT INTO Inserisce (Utente, ValoreEconomico, DataInizio, DataFine)
VALUES
('utente1@example.com', 5.00, '2024-06-01', '2024-06-30'),
('utente2@example.com', 10.00, '2024-06-01', '2024-06-30'),
('utente3@example.com', 15.00, '2024-06-01', '2024-06-30'),
('utente4@example.com', 20.00, '2024-06-01', '2024-06-30'),
('utente5@example.com', 25.00, '2024-06-01', '2024-06-30');
```

Popolamento della tabella Contiene

```
INSERT INTO Contiene (Titolo, DataOrdine, CreazioneOrdine, EmailUtente,
NomeRist, ViaRist, CivicoRist, CAPRist) VALUES
('Pizza Margherita', '2024-06-01', '12:00:00', 'utente1@example.com',
'Ristorante A', 'Via Roma', 1, '00100'),
```

```
('Sushi', '2024-06-01', '13:00:00', 'utente2@example.com', 'Ristorante B', 'Via Milano', 2, '00200'),
('Taco', '2024-06-01', '14:00:00', 'utente3@example.com', 'Ristorante C', 'Via Napoli', 3, '00300'),
('Curry', '2024-06-01', '15:00:00', 'utente4@example.com', 'Ristorante D', 'Via Firenze', 4, '00400'),
('Ramen', '2024-06-01', '16:00:00', 'utente5@example.com', 'Ristorante E', 'Via Venezia', 5, '00500');
```

Popolamento della tabella Composizione

```
INSERT INTO Composizione (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist,
NomeIngrediente, OrigineDiArrivo) VALUES
('Pizza Margherita', 'Ristorante A', 'Via Roma', 1, '00100', 'Pomodoro', 'Italia'),
('Pizza Margherita', 'Ristorante A', 'Via Roma', 1, '00100', 'Mozzarella', 'Italia'),
('Sushi', 'Ristorante B', 'Via Milano', 2, '00200', 'Riso', 'Giappone'),
('Sushi', 'Ristorante B', 'Via Milano', 2, '00200', 'Avocado', 'Messico'),
('Taco', 'Ristorante C', 'Via Napoli', 3, '00300', 'Avocado', 'Messico'),
('Taco', 'Ristorante C', 'Via Napoli', 3, '00300', 'Pollo', 'India'),
('Curry', 'Ristorante D', 'Via Firenze', 4, '00400', 'Pollo', 'India'),
('Curry', 'Ristorante D', 'Via Firenze', 4, '00400', 'Riso', 'Giappone'),
('Ramen', 'Ristorante E', 'Via Venezia', 5, '00500', 'Riso', 'Giappone'),
('Ramen', 'Ristorante E', 'Via Venezia', 5, '00500', 'Pollo', 'India');
```

Popolamento della tabella Presenta

```
INSERT INTO Presenta (Titolo, NomeRist, ViaRist, CivicoRist, CAPRist,
NomeAllergene) VALUES
('Pizza Margherita', 'Ristorante A', 'Via Roma', 1, '00100', 'Glutine'),
('Pizza Margherita', 'Ristorante A', 'Via Roma', 1, '00100', 'Lattosio'),
('Sushi', 'Ristorante B', 'Via Milano', 2, '00200', 'Pesce'),
('Taco', 'Ristorante C', 'Via Napoli', 3, '00300', 'Glutine'),
('Curry', 'Ristorante D', 'Via Firenze', 4, '00400', 'Glutine'),
('Ramen', 'Ristorante E', 'Via Venezia', 5, '00500', 'Glutine');
```

3.3 Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne:

Cancellazione di un Utente con ordini associati
DELETE FROM Utente WHERE Email = 'utente1@example.com';
Commento:
Questa operazione di cancellazione fallisce a causa dei vincoli di chiave esterna con la tabella Ordine.

Cancellazione di un Piatto ordinato
DELETE FROM Piatto WHERE Titolo = 'Sushi' AND NomeRist = 'Ristorante B' AND ViaRist = 'Via Milano' AND CivicoRist = 2 AND CAPRist = '00200';
Commento:
Questa operazione di cancellazione fallisce a causa dei vincoli di chiave esterna con la tabella Contiene.

Modifica del nome di un Ristorante
UPDATE Ristorante SET Nome = 'Ristorante D2' WHERE Nome = 'Ristorante D' AND Via = 'Via Firenze' AND Civico = 4 AND CAP = '00400';
Commento:
Questa operazione di modifica fallisce a causa dei vincoli di chiave esterna con la tabella Ordine.

Modifica del MezzoDiTrasporto per un Fattorino esistente
UPDATE MezzoDiTrasporto SET NumeroDiSerie = 'PG300' WHERE

NumeroDiSerie = 'CD456';

Commento:

Questa operazione di modifica fallisce a causa dei vincoli di chiave esterna con la tabella Fattorino.

Restrizione chiave esterna per eliminazione di elementi in Utente

```
ALTER TABLE Ordine
ADD CONSTRAINT FK_Ordine_Utente FOREIGN KEY (EmailUtente)
REFERENCES Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE Messaggio
ADD CONSTRAINT FK_Messaggio_Utente FOREIGN KEY (EmailUtente)
REFERENCES Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE Reclamo
ADD CONSTRAINT FK_Reclamo_Utente FOREIGN KEY (EmailUtente)
REFERENCES Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE Recensione
ADD CONSTRAINT FK_Recensione_Utente FOREIGN KEY (EmailUtente)
REFERENCES Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE MezzoDiPagamento
ADD CONSTRAINT FK_MezzoDiPagamento_Utente FOREIGN KEY (Utente)
REFERENCES Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE BorsellinoElettronico
ADD CONSTRAINT FK_BorsellinoElettronico_Utente FOREIGN KEY (Utente)
REFERENCES Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE Inserisce
ADD CONSTRAINT FK_Inserisce_Utente FOREIGN KEY (Utente) REFERENCES
Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE Associa
ADD CONSTRAINT FK_Associa_Utente FOREIGN KEY (Utente) REFERENCES
Utente(Email) ON DELETE CASCADE;
```

```
ALTER TABLE Sconto
ADD CONSTRAINT FK_Sconto_Utente FOREIGN KEY (Utente) REFERENCES
Utente(Email) ON DELETE CASCADE;
```

Commento:

Modifica delle tabelle Ordine, Messaggio, Reclamo, Recensione, MezzoDiPagamento, BorsellinoElettronico, Inserisce, Associa e Sconto per permettere la cancellazione a cascata dalla tabella Utente.

3.4 Qualche query per testare il corretto funzionamento del database:

Query per trovare tutti i fattorini

```
SELECT *
FROM Fattorino;
```

Restituzione:

	idfattorino [PK] integer	posizione character varying (100)	stato character varying (50)	numerodiserie character varying (50)	dataaggclassifica date	posizclassifica integer	tipoclassifica character varying (50)
1	1	Zona A	Disponibile	AB123	2024-06-01	1	Fattorino
2	2	Zona B	Disponibile	CD456	2024-06-01	2	Fattorino
3	3	Zona C	Non Disponibile	EF789	2024-06-01	3	Fattorino
4	4	Zona D	Disponibile	GH012	2024-06-01	4	Fattorino
5	5	Zona E	Non Disponibile	IJ345	2024-06-01	5	Fattorino

Query per trovare tutti gli ordini effettuati da uno specifico utente

```
SELECT *
FROM Ordine
WHERE EmailUtente = 'utente2@example.com';
```

Restituzione:

	datacreazione [PK] date	orariocreazione [PK] time without time zone	emailutente [PK] character varying (100)	nomerist [PK] character varying (100)	viarist [PK] character varying (100)	civicorist [PK] integer	caprist [PK] character
1	2024-06-01	13:00:00	utente2@example.com	Ristorante B	Via Milano	2	00200

via character varying (100)	civico integer	cap character	prezzofinale numeric (10,2)	fattorino integer	orariopresacarico time without time zone	orarioconsegna time without time zone	mancia numeric (5,2)
Via Beta	20	00220	30.00	2	13:10:00	13:40:00	3.00

Query per trovare tutti i piatti offerti da uno specifico ristorante

```
SELECT Titolo, Immagine, Prezzo, ListaDiAppartenenza
FROM Piatto
WHERE NomeRist = 'Ristorante C' AND ViaRist = 'Via Napoli' AND CivicoRist = 3 AND CAPRist = '00300';
```

Restituzione:

	titolo character varying (100)	immagine text	prezzo numeric (10,2)	listadiappartenenza character varying (100)
1	Taco	taco.jpg	5.00	Menu Principale

Query per trovare tutti i reclami dei ristoranti

```
SELECT Rec.*
FROM Reclamo Rec JOIN Ristorante Ris ON Rec.NomeRist = Ris.Nome AND
Rec.ViaRist = Ris.Via AND Rec.CivicoRist = Ris.Civico AND Rec.CAPRist =
Ris.CAP;
```

Restituzione:


	dataordine [PK] date	creazioneordine [PK] time without time zone	emailutente [PK] character varying (100)	nomerist [PK] character varying (100)	viarist [PK] character varying (100)
1	2024-06-01	12:00:00	utente1@example.com	Ristorante A	Via Roma
2	2024-06-01	13:00:00	utente2@example.com	Ristorante B	Via Milano
3	2024-06-01	14:00:00	utente3@example.com	Ristorante C	Via Napoli
4	2024-06-01	15:00:00	utente4@example.com	Ristorante D	Via Firenze
5	2024-06-01	16:00:00	utente5@example.com	Ristorante E	Via Venezia

	civicorist [PK] integer	caprist [PK] character	data date	ora time without time zone	testo text
1	00100		2024-06-01	12:45:00	Ordine in ritardo
2	00200		2024-06-01	13:50:00	Ordine sbagliato
3	00300		2024-06-01	14:55:00	Mancava un piatto
4	00400		2024-06-01	15:55:00	Cibo freddo
5	00500		2024-06-01	16:50:00	Qualità scadente

Query per trovare tutti i mezzi di pagamento associati dagli utenti

```
SELECT Mp.TipoDiPagamento
FROM MezzoDiPagamento Mp JOIN Utente U ON Mp.Utente = U.Email;
```



Restituzione:

	tipodipagamento character varying (50) 
1	Carta di Credito
2	PayPal
3	Satispay
4	Carta di Credito
5	Carta di Credito

Query per trovare gli utenti con abbonamento Premium e la loro posizione in classifica

```
SELECT U.Nome, Cl.Posizione
FROM Utente U
LEFT JOIN Classifica Cl ON U.DataAggClassifica = Cl.DataAggiornamento AND
U.PosizClassifica = Cl.Posizione AND U.TipoClassifica = Cl.Tipo
WHERE U.Premium = TRUE;
```

Restituzione:

	nome character varying (100) 	posizione integer 
1	Utente Uno	1
2	Utente Tre	3
3	Utente Cinque	5

Query per trovare i piatti e gli allergeni dei piatti di uno specifico ristorante

```
SELECT Pi.Titolo, Pi.ListaDiAppartenenza, Al.Nome AS Allergene,
Al.Descrizione
FROM Piatto Pi JOIN Presenta Pr ON Pi.Titolo = Pr.Titolo AND Pi.NomeRist =
Pr.NomeRist AND Pi.ViaRist = Pr.ViaRist AND
```

Pi.CivicoRist = Pr.CivicoRist AND Pi.CAPRist = Pr.CAPRist
 JOIN Allergeni Al ON Pr.NomeAllergene = Al.Nome
 WHERE Pi.NomeRist = 'Ristorante E' AND Pi.ViaRist = 'Via Venezia' AND
 Pi.CivicoRist = 5 AND Pi.CAPRist = '00500';

Restituzione:

	titolo character varying (100) 🔒	listadiappartenenza character varying (100) 🔒	allergene character varying (100) 🔒	descrizione text 🔒
1	Ramen	Menu Principale	Glutine	Presente nei cereali