Laboratorio Basi di Dati 2023/2024

Progetto di piattaforma di food delivery CIBORA

Componenti del gruppo: Camerano Alessio (1044444)

Indice

REQUISITI INIZIALI

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

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

L'utente può collezionare codici di sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.

Ogni ristorante (Figura 1(b)) è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un'immagine di profilo e un numero di 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.

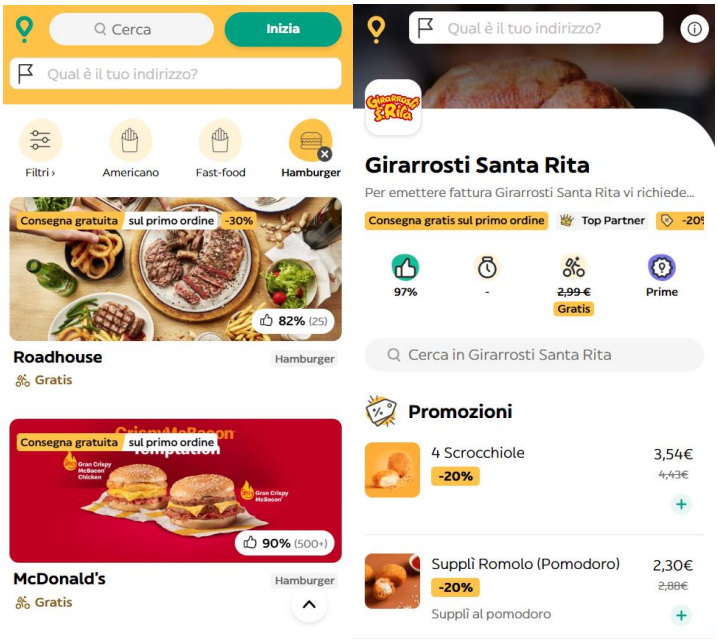


Figura 1 (a) La lista dei ristoranti con filtro “Hamburger”. (b) I dettagli di un ristorante.

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 (bicicletta normale, bicicletta 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

REQUISITI RIVISTI

~~Si deve progettare~~ Dobbiamo 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~~ Gli utenti li rappresentiamo con nome, e-mail, password, numero di telefono, indirizzo di recapito. Una volta registratosi, l'utente deve inserire un mezzo di pagamento (es.: carta di credito, PayPal, satispay) e ricaricare il proprio borsellino elettronico. Il borsellino ha un saldo che viene aggiornato ad ogni ~~ordinazione~~ ordine 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.

Per i codici sconto rappresentiamo il codice che li compone e la data di scadenza. I codici scadono dopo 14 giorni.

Ogni e-mail può essere associata ad un solo account utente e lo stesso vale per il contrario.

Ogni ristorante (Figura 1(b)) ~~è rappresentato da~~ lo rappresentiamo con 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~~ alle portate offerte (ad esempio: fast food, vegetariano, ...).

Per le categorie rappresentiamo il titolo ed il numero di membri.

Non è possibile registrare più ristoranti con lo stesso indirizzo e nome.

I ristoranti che dimostrano di saper garantire un ottimo servizio (almeno 20 ordini consegnati correttamente, una valutazione ~~clienti~~ utenti maggiore o uguale a 4.5 stelline su cinque, una percentuale massima di ordini annullati dal ristorante dell'1.5%, una percentuale massima di ordini con reclami 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~~ degli utenti. Per i Top Partner ~~si vuole tenere~~ teniamo traccia della data in cui sono entrati a far parte della categoria.

I ristoranti propongono agli utenti ~~una lista di piatti~~ un elenco di portate 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~~ portata appartiene ad una o più liste (es. i più venduti, promozioni, dolci, salato, ecc.).

Per ogni lista rappresentiamo il titolo.

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

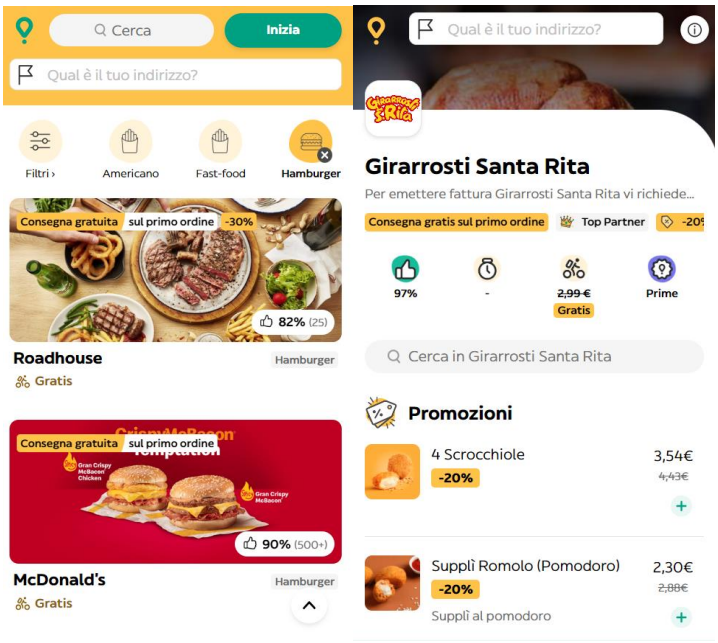


Figura 1 (a) La lista dei ristoranti con filtro "Hamburger". (b) I dettagli di un ristorante.

Il sistema gestisce un numero arbitrario di riders dove ogni rider ~~è identificato da~~ lo identifichiamo con 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 (bicicletta normale, bicicletta 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~~ utente" superiore ai 10 km, solo i rider con bici elettrica vengono interpellati. Per monitorare le prestazioni dei ~~ciclofattorini~~ riders, ~~si vuole tenere~~ teniamo traccia del

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

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

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

Per l'ordine memorizziamo la data e l'orario di creazione, l'orario di consegna e lo status.

Inoltre, per l'utente 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~~ portate più popolari
- Ristoranti con più recensioni positive
- ~~Clienti~~ utenti che hanno speso di più

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Utente	Colui che ha intenzione di effettuare l'ordine sull'applicazione	Cliente	Ristorante Rider Ordine Borsellino CodiciConto
Borsellino	Il saldo dell'utente sull'applicazione usato per effettuare gli ordini		Utente
Rider	Colui che prende in carico l'ordine dell'utente e lo consegna, può usare più tipi di mezzi	Fattorino	Utente Ordine

Ristorante	Offre le portate per gli utenti, può rientrare in più categorie	Ristoratore	Utente Portata Lista Categoria
Portata	Le pietanze offerte dal ristorante hanno una lista di ingredienti ed allergeni	Piatti Pietanza Cibo	Ristorante Lista
Lista	La collezione di portate divise in base a specifiche caratteristiche		Portata Ristorante
Ordine	La richiesta che l'utente fa dall'applicazione al ristorante	Ordinazione	Utente Portata Rider
CodiciSconto	Codici che l'utente ottiene in base a quanti ordini fa e che gli permettono di ottenere sconti		Utente
Categoria	L'insieme delle categorie a cui il ristorante può far parte		Ristorante

Gruppi di frasi omogenee

- **Frasi relative all'utente**
  - Gli utenti li rappresentiamo con nome, e-mail, password, numero di telefono, indirizzo di recapito.
  - Gli utenti possono sottoscrivere la modalità premium che garantisce una priorità sugli ordini.
  - L'utente può ricaricare il proprio borsellino in qualsiasi momento.
  - L'utente può collezionare codici di sconto da utilizzare al momento dell'ordine in base al numero di ordini effettuati in passato.
  - Ogni e-mail può essere associata ad un solo account utente e lo stesso vale per il contrario.
  - Ogni utente può selezionare un elenco di portate ed effettuare l'ordine.

    - 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.
    - Inoltre, per l’utente è anche presente la possibilità di dare una mancia al rider per la consegna.
- **Frasi relative al borsellino**
    - Il borsellino ha un saldo che viene aggiornato ad ogni ordine.
- **Frasi relative al ristorante**
    - Ogni ristorante lo rappresentiamo con 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 alle portate offerte.
    - Non è possibile registrare più ristoranti con lo stesso indirizzo e nome.
    - I ristoranti che dimostrano di saper garantire un ottimo servizio (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.
    - 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 degli utenti. Per i Top Partner teniamo traccia della data in cui sono entrati a far parte della categoria.
    - I ristoranti propongono agli utenti un elenco di portate da ordinare.
    - Del ristorante memorizziamo anche il numero di telefono che daranno a disposizione dell'utente nel momento della creazione dell’ordine. Il ristorante non può cambiare numero di telefono in orario lavorativo.
- **Frasi relative al rider**
    - Ogni rider lo identifichiamo con un codice, dallo stato (occupato/disponibile/fuori servizio), dalla posizione.
    - I riders sono classificati in base al tipo di mezzo che utilizzano.
    - I riders che utilizzano il monopattino devono indicare quanti km possono effettuare prima che si scarichi la batteria.
    - Per monitorare le prestazioni dei riders, teniamo traccia del numero di consegne effettuate da ognuno, del momento in cui l’ordine viene

affidato ad un rider e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato all'utente.
    - Per il rider registriamo anche il numero di telefono messo a disposizione dell'utente durante l'ordinazione, il rider non può cambiare il numero di telefono se è in orario di lavoro.
- **Frasi relative alla portata**
    - Ogni portata ha un titolo, un'immagine, una lista di ingredienti, una lista di allergeni, il prezzo e un eventuale sconto. Inoltre, ogni portata appartiene ad una o più liste.
- **Frasi relative alla lista**
    - Per ogni lista rappresentiamo il titolo.
- **Frasi relative all'ordine**
- Per l'ordine memorizziamo la data e l'orario di creazione, l'orario di consegna, lo status, la mancia per il rider (opzionale).
- **Frasi relative ai CodiciSconto**
    - Per i codici sconto rappresentiamo il codice che li compone e la data di scadenza. I codici scadono dopo 14 giorni.
- **Frasi relative alle categorie**
    - Per le categorie rappresentiamo il titolo ed il numero di membri.

Business Rules

- **Ricavate dal testo**
    - Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dagli utenti, sia dai ristoratori.
    - Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio al ristorante.
    - 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-> utente" superiore ai 10 km, solo i rider con bici elettrica vengono interpellati.
    - I ristoranti che dimostrano di saper garantire un ottimo servizio (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.
    - Una volta al mese, vengono aggiornate le seguenti classifiche:

riders più veloci nel consegnare gli ordini
portate più popolari
ristoranti con più recensioni positive
utenti che hanno speso di più

- Dopo che l'ordine è stato effettuato l'utente, grazie ad un'app di messaggistica, 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.

- **Introdotte e vincoli**
  - Il formato delle e-mail degli utenti è del tipo *text*@gmail.com.
  - L'utente può utilizzare un solo codice sconto durante la creazione dell'ordine
  - I codici sconto scadono dopo 14 giorni dall'ottenimento
  - Ogni rider ha a disposizione un suo mezzo personale per le consegne
  - Ogni e-mail può essere associata ad un solo account utente e lo stesso vale per il contrario.
  - Non è possibile registrare più ristoranti con lo stesso indirizzo e nome.

Schema ER

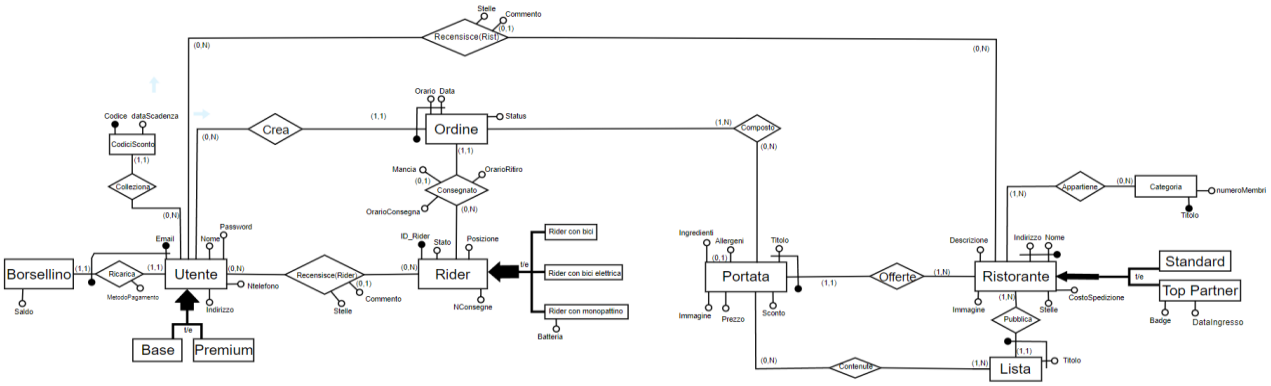


Tavola dei volumi

Concetto	Tipo	Volume	Motivazione
Utente	E	2,5mln	Suppongo che il volume degli utenti

			corrisponda in media a quello degli utenti registrati sull'app Glovo vista la somiglianza del servizio offerto.
Borsellino	E	2.5mln	Ogni utente sull'applicazione avrà il suo borsellino
Rider	E	300k	Suppongo che il numero di riders attivi sia in media valido per gestire l'utenza media.
Ristorante	E	250k	Dati di Unioncamere, suppongo che si registrino sull'applicazione tutti i ristoranti in Italia.
Portata	E	1.5mln	Suppongo che ogni ristorante offra in media 6 piatti per ogni tipologia (antipasto, primo, secondo carne, secondo pesce, contorno, dolce).
Lista	E	1.5mln	Suppongo che ogni ristorante crei circa 6 liste
Ordine	E	5mln	Suppongo che ogni utente effettui in media 2 ordini la settimana e che dopo 7 giorni gli ordini passati vengono rimossi dal database per ottimizzare lo spazio a disposizione

CodiciSconto	E	7.5mln	Metto come limite di codici posseduti nello stesso momento per ogni utente a 3
Categoria	E	1k	Suppongo che l'applicazione raggruppi i ristoranti registrati in 1000 categorie totali per permettere all'utente di cercare più facilmente che cosa cerca in particolare
Utente base	E	1.75mln	Supponendo che gli utenti base siano il 70% del totale
Utente premium	E	750k	Supponendo che gli utenti premium siano il 30% del totale
Rider con bici	E	90k	Supponendo che i rider con bici siano il 30% del totale
Rider con bici elettrica	E	120k	Supponendo che i rider con bici elettrica siano il 40% del totale
Rider con monopattino	E	90k	Supponendo che i rider con monopattino siano il 30% del totale
Ristorante standard	E	200k	Supponendo che i ristoranti standard siano l'80% del totale
Ristorante top partner	E	50k	Supponendo che i ristoranti top

			partner siano il 20% del totale
Ricarica	A	20	Suppongo che in media ricarichi questo ammontare di volte il borsellino
Colleziona	A	100	Suppongo che l'utente collezioni in media questo ammontare di codici all'anno
Crea	A	100	Suppongo che ogni utente crei in media questo ammontare di ordini all'anno
Offre	A	50	Suppongo che ogni ristorante offra in un anno questo ammontare di portate
Appartiene	A	3	Suppongo che Il ristorante appartenga a 3 categorie
Contenuta	A	10	Suppongo che la portata in un anno venga inserita in 10 liste
Pubblica	A	30	Suppongo che un ristorante pubblichi annualmente una 30ina di liste
Consegnato	A	500	Suppongo che un rider annualmente consegni questo ammontare di ordini
composto	A	3	Suppongo che gli ordini siano composti in media da 3 portate
Recensisce (Rider)	A	30	Suppongo che un utente

			annualmente recensisca il rider 30 volte
Recensisce (Rist)	A	30	Suppongo che un utente annualmente recensisca il ristorante 30 volte

Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza
1	Utente si iscrive a Cibora	I	2000 al giorno
2	Utente ricarica il borsellino	I	1 a settimana
3	Ristorante aggiorna il numero di stelle	B	Ogni lunedì
4	Viene aggiornata la posizione del rider tramite GPS	I	Ogni 3 secondi
5	Viene calcolato quale rider verrà selezionato per l'ordine	I	Ogni ordine
6	Viene aggiornata la classifica dei riders più veloci nel consegnare	B	Una volta al mese
7	Viene aggiornata la classifica delle portate più popolari	B	Una volta al mese
8	Viene aggiornata la classifica dei ristoranti con più recensioni positive	B	Una volta al mese
9	Viene aggiornata la classifica degli utenti che han speso di più	B	Una volta al mese

Analisi delle ridondanze

Le ridondanze individuate sono le seguenti:

Analisi 1: data l'operazione "**Per monitorare le prestazioni dei riders, si vuole tenere traccia del numero di consegne effettuate da ognuno**"

Tavola dei volumi necessari

Rider	E	300k
Ordine	E	5mln
Consegnato	A	500

Tavola degli accessi

Operazione 1:

Leggere il numero totale di consegne di un rider (1 volta al giorno)

Ordine	E	1	L
Consegnato	A	1	L
Rider	E	1	L

Senza ridondanza sono 3 accessi al giorno per ogni rider

Rider	E	1	L

Con ridondanza sono 1 accesso al giorno

Ricapitolando:

- **Con ridondanza**
  - Operazione 1:
    - 1 accesso al giorno moltiplicato per numero riders (300k)
  - Totale di 300k **accessi** al giorno

- **Senza ridondanza**
  - Operazione 1:
    - 3 accesso al giorno moltiplicato per numero riders (300k)
  - Totale di 900k **accessi** al giorno

Costo aggiuntivo in termini di spazio (con ridondanza)

Ipotesi: si utilizzano 4 byte per memorizzare il numero di consegne

Spazio totale necessario: 4 * 300k= 1.2mln byte (1200 kbyte)

È conveniente mantenere o eliminare la ridondanza?

Con ridondanza	Senza ridondanza
300k accessi	900k accessi
1200 kbyte di spazio aggiuntivo	0 kbyte di spazio aggiuntivo

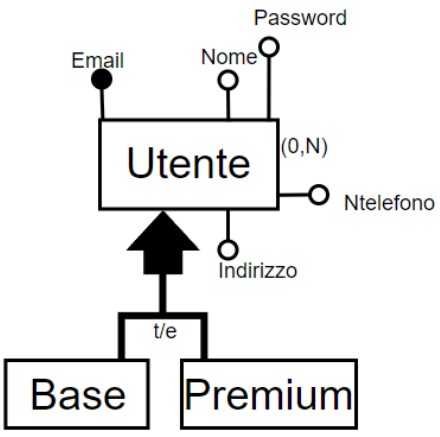
Le analisi dimostrano che conviene tenere la ridondanza perché ho meno accessi da fare e spreco un infimo spazio

Eliminazione delle generalizzazioni

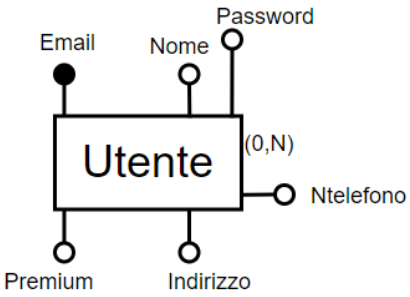
Sullo schema ER iniziale sono state utilizzate 3 generalizzazioni.

Di seguito verrà illustrato come è stato deciso di rimuoverle:

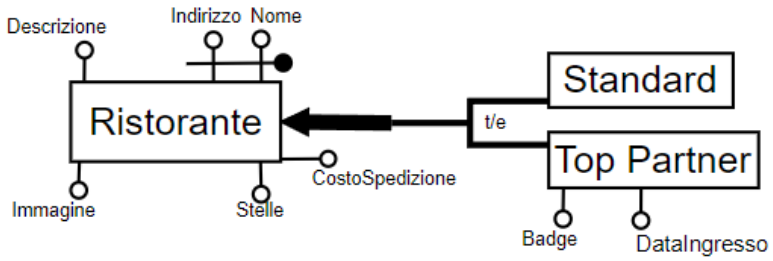
- La prima generalizzazione che viene esaminata è quella che genera l'entità utente (di tipo totale/esclusiva).



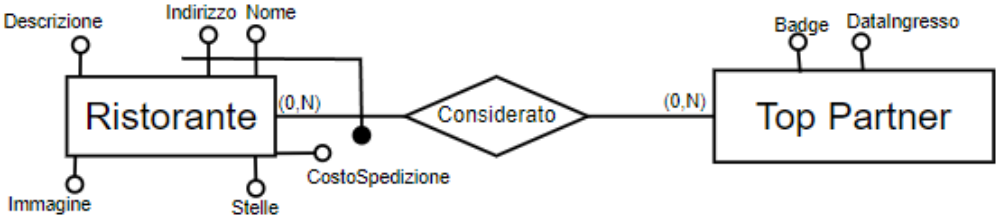
Si è deciso di raggruppare le entità figlie all'interno dell'entità genitore utilizzando l'attributo **premium** per distinguerle:



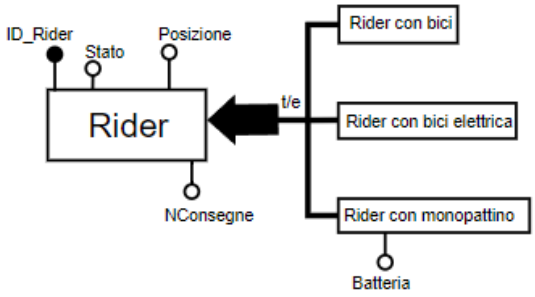
- La seconda generalizzazione che viene analizzata è quella che viene generata dall'entità ristorante (totale/esclusiva).



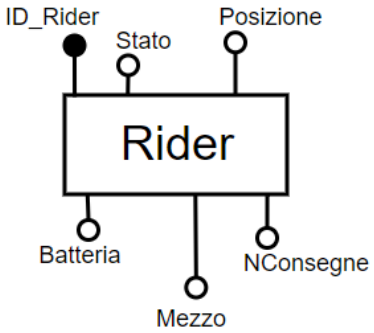
Si è deciso di creare **un'entità a parte** per i Top Partners che verrà collegata poi con Ristorante attraverso l'associazione **considerato**:



- La terza generalizzazione che viene modificata è quella che viene generata dall'entità rider (totale/esclusiva).

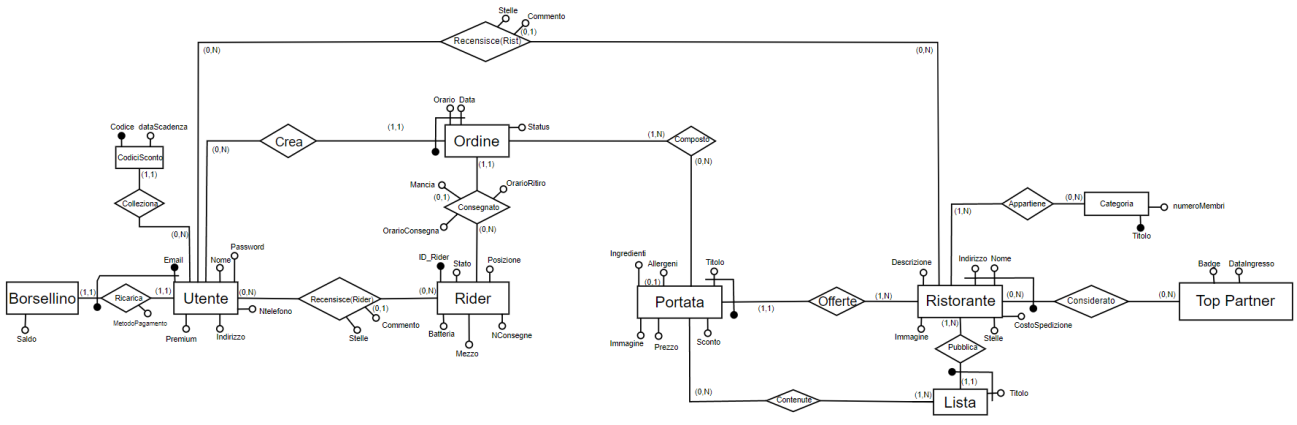


Si è deciso di integrare le entità figlie all'interno dell'entità genitore mettendo come attributi aggiuntivi **mezzo** e **batteria** (attributo che risulterà NULL quando il mezzo sarà diverso da monopattino):



Schema ER ristrutturato

In seguito alle varie modifiche ottenute dall'ottimizzazione logica, lo schema risultato è stato il seguente



Schema Relazionale

- Utente (**e-mail**, nome, password, premium, indirizzo, nTelefono)
- CodiciSconto (**Codice,** dataScadenza)
- Borsellino (**e-mail_utente**, saldo)
    - Borsellino (e-mail_utente) referenzia Utente (e-mail)
- Ricarica (**e-mail_utente, e-mail**, metodoPagamento)
    - Ricarica (e-mail_utente) referenzia Borsellino (e-mail_utente)
    - Ricarica (e-mail) referenzia Utente (e-mail)
- Rider (**ID_Rider**, stato, posizione, mezzo, batteria, nConsegne)
- Recensisce(Rider) (**ID_Rider, e-mail_utente**, stelle, commento*)
    - Recensisce(Rider) (ID_Rider) referenzia Rider (ID_Rider)
    - Recensisce(Rider) (e-mail_utente) referenzia Utente (e-mail)
- Ordine (**e-mail_utente, data, orario**, status)
    - Ordine (e-mail_utente) referenzia Utente (e-mail)
- Consegnato (**e-mail_utente, dataOrdine, orarioOrdine, ID_Rider**, orarioConsegna, mancia*, orarioRitiro)
    - Consegnato (e-mail_utente, dataOrdine, orarioOrdine) referenziano Ordine (e-mail_utente, dataOrdine, orarioOrdine)
    - Consegnato (ID_Rider) referenzia Rider (ID_Rider)
- Ristorante (**nome, indirizzo**, stelle, immagine, costoSpedizione, descrizione)
- Recensisce(Rist) (**e-mail_utente, nomeRistorante, indirizzoRistorante**, stelle, commento*)
    - Recensisce(Rist) (e-mail_utente) referenzia Utente (e-mail)
    - Recensisce(Rist) (nomeRistorante, indirizzoRistorante) referenziano Ristorante (nome, indirizzo)
- Portata (**titolo, nomeRistorante, indirizzoRistorante**, sconto, prezzo, immagine, ingredienti, allergeni*)
    - Portata (nomeRistorante, indirizzoRistorante) referenziano Ristorante (nome, indirizzo)
- Lista (**Titolo, nomeRistorante, indirizzoRistorante**)

    - Lista (nomeRistorante, indirizzoRistorante) referenziano Ristorante (nome, indirizzo)
- Top Partner (**nomeRistorante, indirizzoRistorante**, badge, dataIngresso)
    - Top Partner (nomeRistorante, indirizzoRistorante) referenziano Ristorante (nome, indirizzo)
- Categoria (**Titolo,** numeroMembri)
- Colleziona (**e-mail_utente, CodiceSconto**)
    - Colleziona (e-mail_utente) referenzia Utente (e-mail)
    - Colleziona (codiceSconto) referenzia CodiciSconto (codice)
- Composto (**e-mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante, indirizzoRistorante**)
    - Composto (e-mail_utente, dataOrdine, orarioOrdine) referenziano Ordine (e-mail_utente, data, ora)
    - Composto (nomeRistorante, indirizzoRistorante) referenziano Ristorante (nome, indirizzo)
- Appartiene (**nomeRistorante, indirizzoRistorante, titoloCategoria**)
    - Appartiene (nomeRistorante, IndirizzoRistorante) referenziano Ristorante (nome, indirizzo)
    - Appartiene (titoloCategoria) referenzia Categoria (titolo)

Comandi DDL per la creazione della struttura

Creazione tabella Utente

```
CREATE TABLE Utenti (
email VARCHAR(255) NOT NULL,
nome VARCHAR(100) NOT NULL,
password VARCHAR(20) NOT NULL,
premium BOOLEAN NOT NULL,
indirizzo VARCHAR(255) NOT NULL,
nTelefono VARCHAR(15) NOT NULL ,
primary key(email),
```

```
CHECK (email LIKE '%@gmail.com'),
CHECK (LENGTH(password) BETWEEN 6 AND 20)
)
```

Creazione tabella Borsellino

```
CREATE TABLE Borsellino (
email_utente VARCHAR(255) NOT NULL,
saldo DECIMAL(10, 2) NOT NULL,
primary key(email_utente),
CHECK (saldo >= 0),
FOREIGN KEY (email_utente) REFERENCES Utenti(email)
)
```

Creazione tabella CodiciSconto

```
CREATE TABLE CodiciSconto (
Codice VARCHAR(8) NOT NULL,
DataScadenza DATE NOT NULL,
primary key(Codice),
CHECK (LENGTH(Codice) = 8)
)
```

Creazione tabella Ricarica

```
CREATE TABLE Ricarica (
e_mail_utente VARCHAR(255) NOT NULL,
e_mail VARCHAR(255) NOT NULL,
metodoPagamento VARCHAR(50) NOT NULL,
FOREIGN KEY (e_mail_utente) REFERENCES Borsellino(e_mail_utente),
```

```
FOREIGN KEY (e_mail) REFERENCES Utenti(email),

CHECK (e_mail LIKE '%@gmail.com')

)
```

Creazione tabella Rider

```
CREATE TABLE Rider (

ID_Rider CHAR(8) NOT NULL,

 stato VARCHAR(50) NOT NULL,

posizione VARCHAR(255) NOT NULL,

mezzo VARCHAR(20) NOT NULL,

batteria INT,

numeroConsegne INT,

CHECK (mezzo IN ('bici', 'bici elettrica', 'monopattino'))

CHECK (ID_Rider ~ '^[0-9]{8}$'),

CHECK (batteria BETWEEN 0 AND 100),

CHECK (mezzo <> 'monopattino' OR batteria IS NOT NULL),

primary key (ID_Rider)

)
```

Creazione tabella Ordine

```
CREATE TABLE Ordine (

email_utente VARCHAR(255) NOT NULL,

data DATE NOT NULL,

orario TIME NOT NULL,

status VARCHAR(50) NOT NULL,
```

```
primary key(email_utente, data, orario),
FOREIGN KEY (email_utente) REFERENCES Utente(email)
)
```

Creazione tabella Ristorante

```
CREATE TABLE Ristorante (
nome VARCHAR(100) NOT NULL,
indirizzo VARCHAR(255) NOT NULL,
stelle DECIMAL(2, 1) NOT NULL,
immagine VARCHAR(255) NOT NULL,
costoSpedizione DECIMAL(10, 2) NOT NULL,
descrizione TEXT NOT NULL,
primary key (nome, indirizzo),
CHECK (stelle BETWEEN 1.0 AND 5.0),
CHECK (immagine LIKE '%.jpg'),
CHECK (costoSpedizione >= 0)
)
```

Creazione tabella Portata

```
CREATE TABLE Portata (
titolo VARCHAR(100) NOT NULL,
nomeRistorante VARCHAR(100) NOT NULL,
indirizzoRistorante VARCHAR(255) NOT NULL,
sconto DECIMAL(10, 2) NOT NULL,
prezzo DECIMAL(10, 2) NOT NULL,
```

immagine VARCHAR(255) NOT NULL,

ingredienti TEXT NOT NULL,

allergeni TEXT,

primary key(titolo, nomeRistorante, indirizzoRistorante),

CHECK (sconto >= 0),

CHECK (prezzo >= 0),

CHECK (immagine LIKE '%.jpg'),

FOREIGN KEY (nomeRistorante, indirizzoRistorante) REFERENCES Ristorante(nome, indirizzo)

)

Creazione tabella Lista

CREATE TABLE Lista (

Titolo VARCHAR(100) NOT NULL,

nomeRistorante VARCHAR(100) NOT NULL,

indirizzoRistorante VARCHAR(255) NOT NULL,

primary key (Titolo, nomeRistorante, indirizzoRistorante),

FOREIGN KEY (nomeRistorante, indirizzoRistorante) REFERENCES Ristorante(nome, indirizzo)

)

Creazione tabella Top Partner

CREATE TABLE TopPartner (

nomeRistorante VARCHAR(100) NOT NULL,

indirizzoRistorante VARCHAR(255) NOT NULL,

badge BOOLEAN NOT NULL,

dataIngresso DATE NOT NULL,

primary key (nomeRistorante, indirizzoRistorante),

FOREIGN KEY (nomeRistorante, indirizzoRistorante) REFERENCES Ristorante(nome, indirizzo)

)

Creazione tabella Categoria

CREATE TABLE Categoria (

Titolo VARCHAR(100) NOT NULL,

numeroPartecipanti INT NOT NULL,

primary key(Titolo)

)

Creazione tabella Consegnato

CREATE TABLE Consegnato (

e_mail_utente VARCHAR(255) NOT NULL,

dataOrdine DATE NOT NULL,

orarioOrdine TIME NOT NULL,

ID_Rider INT NOT NULL,

orarioConsegna TIME NOT NULL,

mancia DECIMAL(10, 2) NOT NULL,

orarioRitiro TIME NOT NULL,

CHECK (mancia >= 0),

primary key (e_mail_utente, dataOrdine, orarioOrdine, ID_Rider),

FOREIGN KEY (e_mail_utente, dataOrdine, orarioOrdine) REFERENCES Ordine (e_mail_utente, dataOrdine, orarioOrdine),

FOREIGN KEY (ID_Rider) REFERENCES Rider (ID_Rider)

)

Creazione tabella Recensisce(Rider)

```
CREATE TABLE Recensisce(Rider) (
ID_Rider INT NOT NULL,
e_mail_utente VARCHAR(255) NOT NULL,
stelle INT NOT NULL CHECK (stelle >= 1 AND stelle <= 5),
commento TEXT,
primary key (ID_Rider, e_mail_utente),
FOREIGN KEY (ID_Rider) REFERENCES Rider (ID_Rider),
FOREIGN KEY (e_mail_utente) REFERENCES Utente (e_mail)
)
```

Creazione tabella Recensisce(Rist)

```
CREATE TABLE Recensisce(Rist) (
e_mail_utente VARCHAR(255) NOT NULL,
nomeRistorante VARCHAR(255) NOT NULL,
indirizzoRistorante VARCHAR(255) NOT NULL,
stelle INT NOT NULL,
commento TEXT,
CHECK (stelle >= 1 AND stelle <= 5),
primary key (e_mail_utente, nomeRistorante, indirizzoRistorante),
FOREIGN KEY (e_mail_utente) REFERENCES Utente (e_mail),
FOREIGN KEY (nomeRistorante, indirizzoRistorante)
REFERENCES Ristorante (nome, indirizzo)
)
```

Creazione tabella Colleziona

```
CREATE TABLE Colleziona (

e_mail_utente VARCHAR(255) NOT NULL,

CodiceSconto VARCHAR(8) NOT NULL,

primary key (e_mail_utente, CodiceSconto),

FOREIGN KEY (e_mail_utente) REFERENCES Utenti(email),

FOREIGN KEY (CodiceSconto) REFERENCES CodiciSconto(Codice)

)
```

Creazione tabella Composto

```
CREATE TABLE Composto (

e_mail_utente VARCHAR(255) NOT NULL,

dataOrdine DATE NOT NULL,

orarioOrdine TIME NOT NULL,

titoloPortata VARCHAR(100) NOT NULL,

nomeRistorante VARCHAR(100) NOT NULL,

indirizzoRistorante VARCHAR(255) NOT NULL,

primary key (e_mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante,
indirizzoRistorante),

FOREIGN KEY (e_mail_utente, dataOrdine, orarioOrdine) REFERENCES
Ordine(email_utente, data, orario),

FOREIGN KEY (nomeRistorante, indirizzoRistorante) REFERENCES Ristorante(nome,
indirizzo)

)
```

Creazione tabella appartiene

```
CREATE TABLE Appartiene (

nomeRistorante VARCHAR(100) NOT NULL,
```

```
indirizzoRistorante VARCHAR(255) NOT NULL,

titoloCategoria VARCHAR(100) NOT NULL,

primary key  (nomeRistorante, indirizzoRistorante, titoloCategoria),

FOREIGN KEY (nomeRistorante, indirizzoRistorante) REFERENCES Ristorante(nome, indirizzo),

FOREIGN KEY (titoloCategoria) REFERENCES Categoria(Titolo)

)
```

Comandi DML per il popolamento della base di dati

Utente

```
INSERT INTO Utenti (email, nome, password, premium, indirizzo, nTelefono) VALUES

('user1@gmail.com', 'Mario Rossi', 'password1', TRUE, 'Via Roma 1', '1234567890'),

INSERT INTO Utenti (email, nome, password, premium, indirizzo, nTelefono) VALUES

('user2@gmail.com', 'Luca Bianchi', 'password2', FALSE, 'Via Milano 2', '0987654321'),

INSERT INTO Utenti (email, nome, password, premium, indirizzo, nTelefono) VALUES

('user3@gmail.com', 'Anna Verdi', 'password3', TRUE, 'Via Napoli 3', '1122334455'),

INSERT INTO Utenti (email, nome, password, premium, indirizzo, nTelefono) VALUES

('user4@gmail.com', 'Marco Neri', 'password4', FALSE, 'Via Torino 4', '2233445566'),

INSERT INTO Utenti (email, nome, password, premium, indirizzo, nTelefono) VALUES

('user5@gmail.com', 'Sara Gialli', 'password5', TRUE, 'Via Firenze 5', '3344556677');
```

Borsellino

```
INSERT INTO Borsellino (e_mail_utente, saldo) VALUES

('user1@gmail.com', 100.00),
```

```
INSERT INTO Borsellino (e_mail_utente, saldo) VALUES
('user2@gmail.com', 50.00),
INSERT INTO Borsellino (e_mail_utente, saldo) VALUES
('user3@gmail.com', 75.50),
INSERT INTO Borsellino (e_mail_utente, saldo) VALUES
('user4@gmail.com', 20.00),
INSERT INTO Borsellino (e_mail_utente, saldo) VALUES
('user5@gmail.com', 150.00);
```

CodiciSconto

```
INSERT INTO CodiciSconto (Codice, dataScadenza) VALUES
('ABCDEFGH, 15/06/2024'),
INSERT INTO CodiciSconto (Codice, dataScadenza) VALUES
('IJKLMNOP, 01/07/2024'),
INSERT INTO CodiciSconto (Codice, dataScadenza) VALUES
('QRSTUVWX, 20/07/2024'),
INSERT INTO CodiciSconto (Codice, dataScadenza) VALUES
('YZABCDEF, 05/08/2024'),
INSERT INTO CodiciSconto (Codice, dataScadenza) VALUES
('GHIJKLMN, 25/08/2024');
```

Ricarica

```
INSERT INTO Ricarica (e_mail_utente, e_mail, metodoPagamento) VALUES
('user1@gmail.com', 'user1@gmail.com', 'Carta di Credito'),
INSERT INTO Ricarica (e_mail_utente, e_mail, metodoPagamento) VALUES
('user2@gmail.com', 'user2@gmail.com', 'PayPal'),
INSERT INTO Ricarica (e_mail_utente, e_mail, metodoPagamento) VALUES
('user3@gmail.com', 'user3@gmail.com', 'Bonifico Bancario'),
```

```
INSERT INTO Ricarica (e_mail_utente, e_mail, metodoPagamento) VALUES
('user4@gmail.com', 'user4@gmail.com', 'Carta di Credito'),
INSERT INTO Ricarica (e_mail_utente, e_mail, metodoPagamento) VALUES
('user5@gmail.com', 'user5@gmail.com', 'PayPal');
```

Rider

```
INSERT INTO Rider (ID_Rider, stato, posizione, mezzo, batteria, numeroConsegne) VALUES
('12345678', 'Disponibile', 'Via Roma 1', 'bici', NULL, 34),
INSERT INTO Rider (ID_Rider, stato, posizione, mezzo, batteria, numeroConsegne) VALUES
('87654321', 'Occupato', 'Via Milano 2', 'bici elettrica', NULL, 67),
INSERT INTO Rider (ID_Rider, stato, posizione, mezzo, batteria, numeroConsegne) VALUES
('11223344', 'Disponibile', 'Via Napoli 3', 'monopattino', 50, 12),
INSERT INTO Rider (ID_Rider, stato, posizione, mezzo, batteria, numeroConsegne) VALUES
('44332211', 'In pausa', 'Via Torino 4', 'bici', NULL, 54),
INSERT INTO Rider (ID_Rider, stato, posizione, mezzo, batteria, numeroConsegne) VALUES
('55667788', 'Disponibile', 'Via Firenze 5', 'bici elettrica', NULL, 23);
```

Ordine

```
INSERT INTO Ordine (email_utente, data, orario, status) VALUES
('user1@gmail.com', '2024-06-01', '12:30:00', 'Consegnato'),
INSERT INTO Ordine (email_utente, data, orario, status) VALUES
('user2@gmail.com', '2024-06-02', '13:00:00', 'In Preparazione'),
INSERT INTO Ordine (email_utente, data, orario, status) VALUES
('user3@gmail.com', '2024-06-03', '13:30:00', 'Spedito'),
INSERT INTO Ordine (email_utente, data, orario, status) VALUES
('user4@gmail.com', '2024-06-04', '14:00:00', 'Consegnato'),
INSERT INTO Ordine (email_utente, data, orario, status) VALUES
('user5@gmail.com', '2024-06-05', '14:30:00', 'Annullato');
```

Ristorante

INSERT INTO Ristorante (nome, indirizzo, stelle, immagine, costoSpedizione, descrizione) VALUES

('Ristorante A', 'Via Roma 1', 4.5, 'ristoranteA.jpg', 2.50, 'Ristorante italiano'),

INSERT INTO Ristorante (nome, indirizzo, stelle, immagine, costoSpedizione, descrizione) VALUES

('Ristorante B', 'Via Milano 2', 4.0, 'ristoranteB.jpg', 3.00, 'Ristorante cinese'),

INSERT INTO Ristorante (nome, indirizzo, stelle, immagine, costoSpedizione, descrizione) VALUES

('Ristorante C', 'Via Napoli 3', 3.5, 'ristoranteC.jpg', 1.50, 'Ristorante messicano'),

INSERT INTO Ristorante (nome, indirizzo, stelle, immagine, costoSpedizione, descrizione) VALUES

('Ristorante D', 'Via Torino 4', 4.7, 'ristoranteD.jpg', 2.00, 'Ristorante giapponese'),

INSERT INTO Ristorante (nome, indirizzo, stelle, immagine, costoSpedizione, descrizione) VALUES

('Ristorante E', 'Via Firenze 5', 4.3, 'ristoranteE.jpg', 2.80, 'Ristorante indiano');

Portata

INSERT INTO Portata (titolo, nomeRistorante, indirizzoRistorante, sconto, prezzo, immagine, ingredienti, allergeni) VALUES

('Pizza Margherita', 'Ristorante A', 'Via Roma 1', 0.00, 8.50, 'pizza_margherita.jpg', 'Pomodoro, Mozzarella, Basilico', 'Latticini'),

INSERT INTO Portata (titolo, nomeRistorante, indirizzoRistorante, sconto, prezzo, immagine, ingredienti, allergeni) VALUES

('Spaghetti Carbonara', 'Ristorante A', 'Via Roma 1', 0.00, 10.00, 'spaghetti_carbonara.jpg', 'Spaghetti, Uova, Pancetta, Pecorino', 'Glutine, Uova, Latticini'),

INSERT INTO Portata (titolo, nomeRistorante, indirizzoRistorante, sconto, prezzo, immagine, ingredienti, allergeni) VALUES

('Pollo Kung Pao', 'Ristorante B', 'Via Milano 2', 0.00, 9.50, 'pollo_kung_pao.jpg', 'Pollo, Peperoni, Anacardi', 'Frutta a guscio'),

INSERT INTO Portata (titolo, nomeRistorante, indirizzoRistorante, sconto, prezzo, immagine, ingredienti, allergeni) VALUES

('Tacos', 'Ristorante C', 'Via Napoli 3', 0.00, 7.00, 'tacos.jpg', 'Tortilla, Carne, Verdure', 'Glutine'),

INSERT INTO Portata (titolo, nomeRistorante, indirizzoRistorante, sconto, prezzo, immagine, ingredienti, allergeni) VALUES

('Sushi Misto', 'Ristorante D', 'Via Torino 4', 0.00, 15.00, 'sushi_misto.jpg', 'Pesce, Riso, Alga', 'Pesce');

Lista

INSERT INTO Lista (Titolo, nomeRistorante, indirizzoRistorante) VALUES

('Pizze', 'Ristorante A', 'Via Roma 1'),

INSERT INTO Lista (Titolo, nomeRistorante, indirizzoRistorante) VALUES

('Primi', 'Ristorante A', 'Via Roma 1'),

INSERT INTO Lista (Titolo, nomeRistorante, indirizzoRistorante) VALUES

('Secondi pollo', 'Ristorante B', 'Via Milano 2'),

INSERT INTO Lista (Titolo, nomeRistorante, indirizzoRistorante) VALUES

('Tacos', 'Ristorante C', 'Via Napoli 3'),

INSERT INTO Lista (Titolo, nomeRistorante, indirizzoRistorante) VALUES

('Sushi', 'Ristorante D', 'Via Torino 4');

Top Partner

INSERT INTO TopPartner (nomeRistorante, indirizzoRistorante, badge, dataIngresso) VALUES

('Ristorante A', 'Via Roma 1', TRUE, '2023-01-01'),

INSERT INTO TopPartner (nomeRistorante, indirizzoRistorante, badge, dataIngresso) VALUES

('Ristorante B', 'Via Milano 2', TRUE, '2023-02-01'),

INSERT INTO TopPartner (nomeRistorante, indirizzoRistorante, badge, dataIngresso) VALUES

('Ristorante C', 'Via Napoli 3', FALSE, '2023-03-01'),

INSERT INTO TopPartner (nomeRistorante, indirizzoRistorante, badge, dataIngresso) VALUES

('Ristorante D', 'Via Torino 4', TRUE, '2023-04-01'),

INSERT INTO TopPartner (nomeRistorante, indirizzoRistorante, badge, dataIngresso) VALUES

('Ristorante E', 'Via Firenze 5', FALSE, '2023-05-01');

Categoria

INSERT INTO Categoria (Titolo) VALUES

('Italiano', 123),

INSERT INTO Categoria (Titolo) VALUES

('Cinese', 256),

INSERT INTO Categoria (Titolo) VALUES

('Messicano', 349),

INSERT INTO Categoria (Titolo) VALUES

('Vegetariano', 581),

INSERT INTO Categoria (Titolo) VALUES

('Indiano', 435);

Consegnato

INSERT INTO Consegnato (e_mail_utente, dataOrdine, orarioOrdine, ID_Rider, orarioConsegna, mancia, orarioRitiro) VALUES

('user1@gmail.com', '2024-06-01', '12:30:00', '12345678', '13:00:00', 5.00, '12:45:00'),

INSERT INTO Consegnato (e_mail_utente, dataOrdine, orarioOrdine, ID_Rider, orarioConsegna, mancia, orarioRitiro) VALUES

('user2@gmail.com', '2024-06-02', '13:00:00', '87654321', '13:30:00', 3.00, '13:15:00'),

INSERT INTO Consegnato (e_mail_utente, dataOrdine, orarioOrdine, ID_Rider, orarioConsegna, mancia, orarioRitiro) VALUES

('user3@gmail.com', '2024-06-03', '13:30:00', '11223344', '14:00:00', 4.00, '13:45:00'),

INSERT INTO Consegnato (e_mail_utente, dataOrdine, orarioOrdine, ID_Rider, orarioConsegna, mancia, orarioRitiro) VALUES

('user4@gmail.com', '2024-06-04', '14:00:00', '44332211', '14:30:00', 2.00, '14:15:00'),

INSERT INTO Consegnato (e_mail_utente, dataOrdine, orarioOrdine, ID_Rider, orarioConsegna, mancia, orarioRitiro) VALUES

('user5@gmail.com', '2024-06-05', '14:30:00', '55667788', '15:00:00', 6.00, '14:45:00');

Recensisce(Rider)

INSERT INTO RecensisceRider (ID_Rider, e_mail_utente, stelle, commento) VALUES

('12345678', 'user1@gmail.com', 5, 'Ottimo servizio'),

INSERT INTO RecensisceRider (ID_Rider, e_mail_utente, stelle, commento) VALUES

('87654321', 'user2@gmail.com', 4, 'Puntuale e cortese'),

INSERT INTO RecensisceRider (ID_Rider, e_mail_utente, stelle, commento) VALUES

('11223344', 'user3@gmail.com', 3, 'Servizio nella media'),

INSERT INTO RecensisceRider (ID_Rider, e_mail_utente, stelle, commento) VALUES

('44332211', 'user4@gmail.com', 5, 'Molto soddisfatto'),

INSERT INTO RecensisceRider (ID_Rider, e_mail_utente, stelle, commento) VALUES

('55667788', 'user5@gmail.com', 2, ‘Deludente');

Recensisce(Rist)

INSERT INTO RecensisceRist (e_mail_utente, nomeRistorante, indirizzoRistorante, stelle, commento) VALUES

('user1@gmail.com', 'Ristorante A', 'Via Roma 1', 5, 'Ottimo cibo e servizio'),

INSERT INTO RecensisceRist (e_mail_utente, nomeRistorante, indirizzoRistorante, stelle, commento) VALUES

('user2@gmail.com', 'Ristorante B', 'Via Milano 2', 4, 'Buona qualità del cibo'),

INSERT INTO RecensisceRist (e_mail_utente, nomeRistorante, indirizzoRistorante, stelle, commento) VALUES

('user3@gmail.com', 'Ristorante C', 'Via Napoli 3', 3, 'Servizio un po' lento'),

INSERT INTO RecensisceRist (e_mail_utente, nomeRistorante, indirizzoRistorante, stelle, commento) VALUES

('user4@gmail.com', 'Ristorante D', 'Via Torino 4', 5),

INSERT INTO RecensisceRist (e_mail_utente, nomeRistorante, indirizzoRistorante, stelle, commento) VALUES

('user5@gmail.com', 'Ristorante E', 'Via Firenze 5', 4, 'Molto soddisfatto, tornerò sicuramente');

Colleziona

INSERT INTO Colleziona (e_mail_utente, CodiceSconto) VALUES

('user1@gmail.com', 'ABCDEFGH'),

INSERT INTO Colleziona (e_mail_utente, CodiceSconto) VALUES

('user2@gmail.com', 'IJKLMNOP'),

INSERT INTO Colleziona (e_mail_utente, CodiceSconto) VALUES

('user3@gmail.com', 'QRSTUVWX'),

INSERT INTO Colleziona (e_mail_utente, CodiceSconto) VALUES

('user4@gmail.com', 'YZABCDEF'),

INSERT INTO Colleziona (e_mail_utente, CodiceSconto) VALUES

('user5@gmail.com', 'GHIJKLMN');

Composto

INSERT INTO Composto (e_mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante, indirizzoRistorante) VALUES

('user1@gmail.com', '2023-06-01', '12:00:00', 'Pizza Margherita', 'Ristorante A', 'Via Roma 1'),

INSERT INTO Composto (e_mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante, indirizzoRistorante) VALUES

('user2@gmail.com', '2023-06-02', '13:00:00', 'Spaghetti Carbonara', 'Ristorante B', 'Via Milano 2'),

INSERT INTO Composto (e_mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante, indirizzoRistorante) VALUES

('user3@gmail.com', '2023-06-03', '14:00:00', 'Lasagna', 'Ristorante C', 'Via Napoli 3'),

INSERT INTO Composto (e_mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante, indirizzoRistorante) VALUES

('user4@gmail.com', '2023-06-04', '15:00:00', 'Risotto alla Milanese', 'Ristorante D', 'Via Torino 4'),

INSERT INTO Composto (e_mail_utente, dataOrdine, orarioOrdine, titoloPortata, nomeRistorante, indirizzoRistorante) VALUES

('user5@gmail.com', '2023-06-05', '16:00:00', 'Tiramisu', 'Ristorante E', 'Via Firenze 5');

Appartiene

INSERT INTO Appartiene (nomeRistorante, indirizzoRistorante, titoloCategoria) VALUES

('Ristorante A', 'Via Roma 1', 'Italiano'),

INSERT INTO Appartiene (nomeRistorante, indirizzoRistorante, titoloCategoria) VALUES

('Ristorante B', 'Via Milano 2', 'Giapponese'),

INSERT INTO Appartiene (nomeRistorante, indirizzoRistorante, titoloCategoria) VALUES

('Ristorante C', 'Via Napoli 3', 'Messicano'),

INSERT INTO Appartiene (nomeRistorante, indirizzoRistorante, titoloCategoria) VALUES

('Ristorante D', 'Via Torino 4', 'Cinese'),

INSERT INTO Appartiene (nomeRistorante, indirizzoRistorante, titoloCategoria) VALUES

('Ristorante E', 'Via Firenze 5', 'Francese');

Operazione di cancellazione e modifica per la verifica dei vincoli

- DELETE FROM Top Partner WHERE TopPartner.badge = FALSE;
    - In questo caso se il badge = false non possiamo aggiungere una data di ingresso nei top Partners.
- DELETE FROM Utenti WHERE email = 'user1@gmail.com';
    - Questo tentativo di cancellazione fallirà se ci sono record in altre tabelle (ad es. Borsellino, Colleziona, RecensisceRist, RecensisceRider, Composto) che fanno riferimento a user1@gmail.com. I vincoli di chiave esterna impediscono la cancellazione di un utente se esistono record dipendenti.
- UPDATE Ristorante SET nome = 'Ristorante Z' WHERE nome = 'Ristorante A' AND indirizzo = 'Via Roma 1';
    - Questo comando modificherà il nome del ristorante 'Ristorante A' in 'Ristorante Z'. La modifica sarà possibile solo se i vincoli di chiave esterna nelle tabelle che referenziano Ristorante (ad es. Composto, Appartiene, RecensisceRist) sono aggiornati automaticamente o se non esistono record dipendenti.
- INSERT INTO Colleziona (e_mail_utente, CodiceSconto) VALUES ('nonexistent@gmail.com', 'ABCDEFGH');
    - Questo inserimento fallirà perché l'email 'nonexistent@gmail.com' non esiste nella tabella Utenti. Il vincolo di chiave esterna garantisce che solo utenti esistenti possano essere referenziati.