

Laboratorio Basi di Dati 2022/2023

Progetto di piattaforma di food delivery

Simone Marengo 1048699

Umberto Rastello 996575

Sommario

1 PROGETTAZIONE CONCETTUALE	2
1.1 Requisiti iniziali	2
1.2 Glossario dei termini	4
1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee	6
1.4 Schema E-R principale + business rules	9
2 PROGETTAZIONE LOGICA.....	10
2.1 Tavola dei Volumi	10
2.2 Tavola delle operazioni	12
2.3 Ristrutturazione dello schema E-R.....	14
2.3.1 Analisi delle Ridondanze.....	14
2.3.2 Eliminazione delle generalizzazioni	18
2.3.3 Eventuale partizionamento/accorpamento di entità e associazioni	18
2.3.4 Eventuale eliminazione degli attributi composti e degli attributi multivalore	19
2.3.5 Eventuale scelta degli identificatori principali	19
2.4 Schema E-R ristrutturato + business rules	20
2.5 Schema Relazionale	21
3.0 IMPLEMENTAZIONE	23
3.1 DDL di creazione del Database	23
3.2 DML di popolamento di tutte le tabelle del Database	31
3.3 Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne.....	34

1 PROGETTAZIONE CONCETTUALE

1.1 Requisiti iniziali

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

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

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

Ogni utente può selezionare una lista di pietanze ed effettuare l'ordine. Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dai clienti, sia dai ristoratori. Nel profilo dell'utente si possono ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio al ristorante. Figura 1 (a) La lista dei ristoranti con filtro "Hamburger". (b) I dettagli di un ristorante.

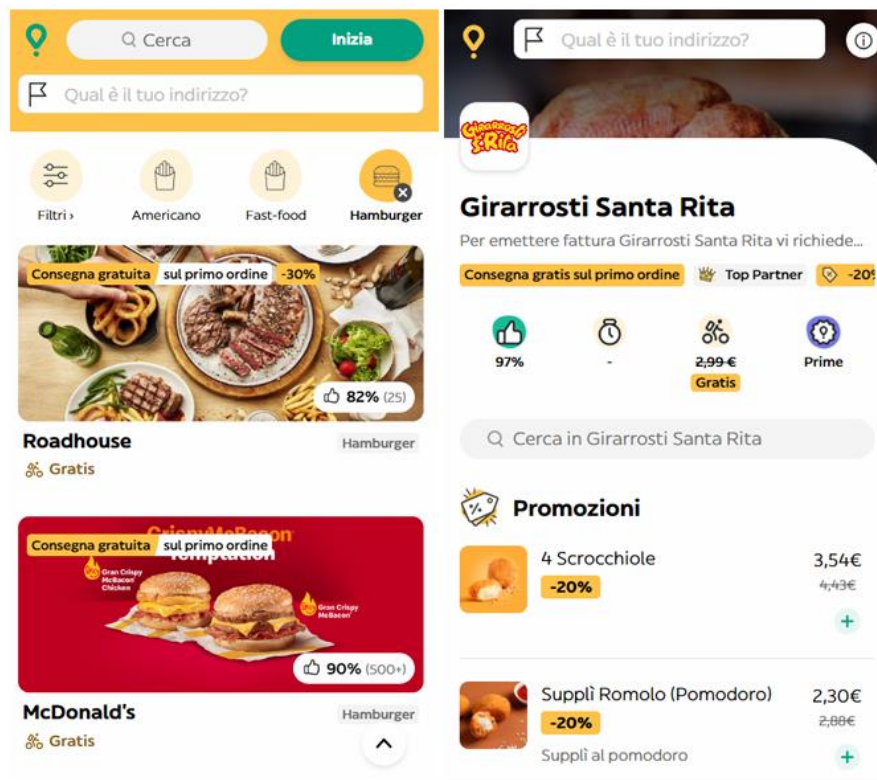


Figura 1 (a) La lista dei ristoranti con filtro "Hamburger". (b) I dettagli di un ristorante.

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

1.2 Glossario dei termini

termine	descrizione	sinonimo	collegamento
Cibora	Il nome del servizio di food delivery per cui si sta progettando la base di dati.	App (mobile), Sistema, Servizio (inteso come Servizio applicativo)	
Utente	Chiunque si registra sulla piattaforma per ordinare cibo.	Cliente	Modalità di pagamento, Borsellino Elettronico, Ordine, Chat
Borsellino Elettronico	Un sistema di pagamento elettronico utilizzato dagli utenti di Cibora per pagare gli ordini di cibo. Il saldo può essere ricaricato in qualsiasi momento.	Saldo	Utente
Modalità Premium	Un'opzione di abbonamento per gli utenti che offre vantaggi come la priorità sugli ordini.		Utente
Codici Sconto	Benefici che gli utenti possono guadagnare e utilizzare per ottenere sconti sugli ordini, basati sul numero di ordini effettuati in passato.		Borsellino Elettronico
Ristorante	Un'entità che prepara e offre cibo per essere consegnato tramite Cibora. Ogni ristorante è classificato per categoria di cibo e può avere il riconoscimento di Top Partner.	Ristoratore	Ordine, Piatto, Top Partner, Categoria, Chat
Top Partner	Un riconoscimento assegnato ai ristoranti che soddisfano criteri specifici di prestazione e servizio.		Ristorante
Ordine	Un acquisto di una o più pietanze fatte dall'utente tramite la piattaforma Cibora. Gli ordini possono essere modificati o annullati (sia dall'utente che dal ristorante) fino a quando non sono stati presi in carico da un rider.	Ordinazione	Utente, Rider, Ristorante, Piatto, Chat

Rider	I fattorini che consegnano gli ordini agli utenti. Possono utilizzare diversi tipi di mezzi come biciclette normali, elettriche o monopattini.	Fattorini, Ciclofattorini	Ordine, Chat
Reclami	Segnalazioni ufficiali fatte dagli utenti relative a problemi incontrati con i servizi offerti dalla piattaforma Cibora		Ordine
Piatti	Le diverse opzioni di cibo offerte dai ristoranti, che possono includere dettagli come ingredienti, allergeni, prezzo e sconti.	Portata, Pietanze, Cibo	Ordine, Ristorante, Lista
Liste	Insieme di piatti raggruppati per categorie (es. i più venduti, promozioni, dolci, salato, ecc.)		Piatto
Consegne	Spostamento di un Piatto dal ristorante all'utente da parte di un rider.		
Chat	Mezzo di comunicazione in tempo reale offerto agli utenti per interagire direttamente con i ristoranti o i rider		Ordine, Rider, Ristorante, Utente
Recensioni	Valutazioni lasciate dagli utenti riguardanti la qualità del servizio del ristorante e del rider, su una scala da 1 a 5.		Ordine
Mancia	Un pagamento opzionale che gli utenti possono dare ai rider come gratifica per il servizio di consegna.		
Classifiche	Liste aggiornate periodicamente che evidenziano i riders più veloci, i cibi più popolari, i ristoranti con più recensioni positive e i clienti che hanno speso di più.		

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

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~~ **rider** che effettuano le consegne in bicicletta.

Per beneficiare ~~del servizio~~ **di Cibora**, 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 **a scelta tra carta di credito, paypal e satispay** (~~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~~ **L'utente può** sottoscrivere la modalità premium **diventando utente premium** che garantisce una priorità sugli ordini. L'utente può collezionare **diversi tipi di** codici di sconto (**aggiunti al proprio borsellino**) ~~da utilizzare al momento dell'ordine~~ in base al numero di ordini effettuati in passato.

Ogni ristorante (Figura 1(b)) è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un'immagine di profilo e un numero di stelline aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al ~~tipo di cibo~~ **piatto** offerto (ad esempio: fast food, vegetariano, ...).

I ristoranti che ~~dimostrano di saper garantire~~ **garantiscono** 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, che compaiono in sezioni dedicate all'interno dell'app mobile di 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.~~ **aumentando** la fiducia degli utenti. Per i Top Partner si vuole tenere traccia della data in cui sono entrati a ~~far~~ **farne parte della** categoria.

I ristoranti propongono agli utenti ~~una lista~~ **un insieme** di piatti ~~da ordinare~~. Ogni ~~portata~~ **piatto** 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~~ **un elenco** di ~~pianze~~ **piatti** ed effettuare l'ordine. **Gli ordini vengono identificati con un id incrementale e hanno una data, un'ora e un prezzo totale.** Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia ~~dai clienti~~ **dagli utenti** che ~~dai ristoranti.~~ **dal ristorante.** ~~Nel profilo dell'utente si possono~~ L'utente può ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio al ristorante.

~~Il sistema~~ **Cibora** 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~~ **Cibora** 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~~ **i riders**, si vuole tenere traccia del numero di consegne effettuate da ~~ognuno~~ **ogni rider**, del momento in cui ~~il cibo~~ **l'ordine** 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~~ **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.~~

L'utente ha la possibilità di comunicare tramite la chat sia con il ristorante che con il rider, in caso di problemi con l'ordine. Le chat hanno identificatori incrementali e salvano ogni singolo messaggio (con data e ora).

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~~ **l'utente ha la possibilità di dare una mancia al rider per la consegna (non è salvata all'interno del DB).**

Una volta al mese, vengono aggiornate le seguenti classifiche:

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

Frase di carattere generale:

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

Frase relative alla fase di registrazione e modalità di abbonamento:

Per beneficiare di Cibora, ogni utente deve registrarsi inserendo nome, e-mail, password, numero di telefono, indirizzo di recapito. Una volta registratosi, l'utente deve inserire un mezzo di pagamento a scelta tra carta di credito, paypal e satispay e ricaricare il proprio borsellino elettronico.

Frase relative alla modalità di abbonamento:

L'utente può sottoscrivere la modalità premium diventando utente premium che garantisce una priorità sugli ordini.

Frase relative ai Ristoranti:

Ogni ristorante (Figura 1(b)) è rappresentato da un nome, una descrizione, un indirizzo, il costo della spedizione, un'immagine di profilo e un numero di stelline aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell'ultima settimana. Ogni ristorante appartiene a una o più categorie in base al piatto offerto (ad esempio: fast food, vegetariano, ...).

I ristoranti che garantiscono 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 che compaiono all'interno di Cibora aumentando la fiducia degli utenti. Per i Top Partner si vuole tenere traccia della data in cui sono entrati a farne parte.

Frase relative ai Piatti:

I ristoranti propongono agli utenti un insieme di piatti. Ogni piatto 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.).

Frase relative agli Utenti e alle Chat:

L'utente può collezionare diversi tipi di codici di sconto (aggiunti al proprio borsellino) in base al numero di ordini effettuati in passato.

Ogni utente può selezionare un elenco di piatti ed effettuare l'ordine. Finché non sono affidati ad un rider per la consegna, gli ordini possono essere annullati sia dagli utenti che dai ristoranti. L'utente può ispezionare gli ordini passati ed eventualmente effettuare dei reclami inviando un messaggio al ristorante.

L'utente ha la possibilità di comunicare tramite la chat sia con il ristorante che con il rider, in caso di problemi con l'ordine. Le chat hanno identificatori incrementali e salvano ogni singolo messaggio (con data e ora).

Frase relative ai Riders e agli ordini che gli vengono assegnati:

Gli ordini vengono identificati con un id incrementale e hanno una data, un'ora e un prezzo totale.

Cibora 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, 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, Cibora 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. Per monitorare i riders, si vuole tenere traccia del numero di consegne effettuate da ogni rider del momento in cui l'ordine da consegnare viene affidato ad un rider e, per le consegne già completate, anche dell'ora in cui l'ordine è stato recapitato all'utente.

Frase relative alle Recensioni:

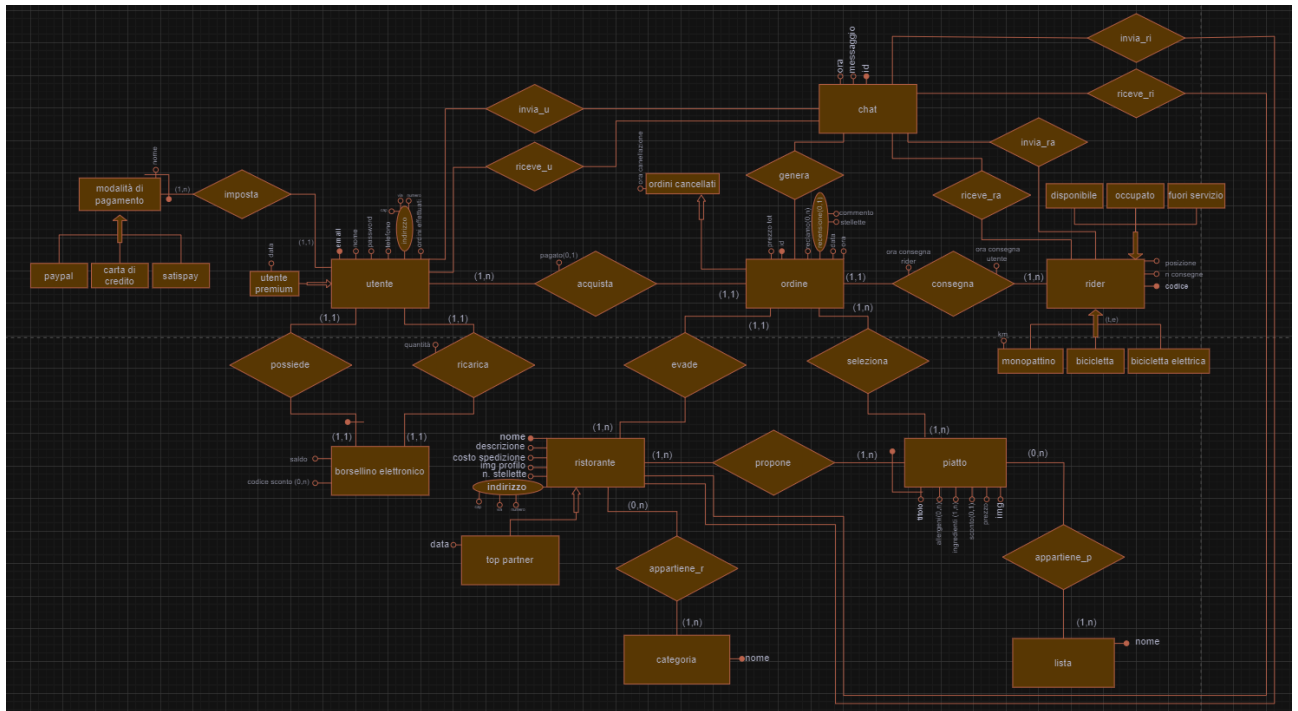
Quando l'ordine è consegnato l'utente può recensire il ristorante e il rider con una valutazione da 1 a 5 e un commento testuale. Il commento testuale è facoltativo. Inoltre, l'utente ha la possibilità di dare una mancia al rider per la consegna (non è salvata all'interno del DB).

Operazioni sui dati:

Una volta al mese, vengono aggiornate le seguenti classifiche:

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

1.4 Schema E-R principale + business rules



//BUSINESS RULES:

Integrità:

- 1- Borsellino Elettronico.Saldo aggiornato ad ogni ordine ($\text{Saldo} = \text{Saldo} - \text{Ordine.PrezzoTot}$)
- 2-Per poter effettuare un ordine, $\text{Borsellino Elettronico.Saldo} \geq \text{Ordine.PrezzoTot}$
- 3-La chat viene creata tra solamente due entità contemporaneamente (utente-ristorante, utente-rider, ristorante-rider)
- 4- $\text{Ristorante.nStellette} = \text{somma voti ultima settimana} / \text{n valutazioni}$, aggiornato ogni settimana
- 5-I ristoranti con queste caratteristiche sono Top Partner: almeno 20 ordini consegnati correttamente, “n stellette” ≥ 4.5 , una percentuale massima di ordini annullati dal ristorante dell’1.5%, una percentuale massima di ordini con reclami del 2.5%)
- 6-Il pagamento dell'ordine consente all'utente di:
effettuare reclami, chattare con il ristorante e con il rider, inserire una recensione. Per effettuare ordine $\text{Acquista.Pagato} = \text{TRUE}$
- 7-Al momento dell'ordine, il sistema Cibora 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.
- 8-Ci sono due tipi di sconti:
il primo è quello che il ristorante imposta sul piatto (Piatto.Sconto),
il secondo tipo di sconto ($\text{Borsellino.CodiciSconto}$), invece, viene ottenuto dal cliente in base al numero e all'importo degli ordini passati.
-dopo il quinto ordine $\gg 20$ per cento
-se l'utente spende più di 20 euro, riceve uno sconto del 15%
-se l'utente effettua almeno un ordine a settimana per 4 settimane, ottiene uno sconto del 50%.
Sarà cura dell'applicativo mostrare i codici sconto posseduti dall'utente
- 9-Se il piatto appartiene ad una determinata lista y, e il ristorante propone quel piatto, allora il ristorante entrerà a far parte di una categoria x legata alla lista y.

Derivazione:

- Utente.OrdiniEffettuati è derivabile dalla partecipazione di Ordini in Consegna
- Rider.nConsegne è derivabile dalla partecipazione di Rider in Consegna
- Ordine.PrezzoTot è derivabile dalla somma di ogni Piatto.Prezzo nell'Ordine

2 PROGETTAZIONE LOGICA

2.1 Tavola dei Volumi

CONCETTO	TIPO	VOLUME
UTENTE	E	34 000 000
UTENTE PREMIUM	E	1 000 000
BORSELLINO ELETTRONICO	E	34 000 000
RISTORANTE	E	48 570
TOP PARTNER	E	8 000
CATEGORIA	E	100
PIATTO	E	50 000
LISTA	E	50
ORDINE	E	500 000 000
ORDINI CANCELLATI	E	10 000 000
RIDER	E	68 000
RIDER_MONOPATTINO	E	8 000

RIDER_BICICLETTA	E	20 000
RIDER_BICICLETTA_E	E	40 000
CHAT	E	1 250 000 000
INVIA_RI	A	12 142 500 000 000
RICEVE_RI	A	12 142 500 000 000
INVIA_RA	A	17 000 000 000 000
RICEVE_RA	A	17 000 000 000 000
INVIA_U	A	850 000 000 000 000
RICEVE_U	A	850 000 000 000 000
IMPOSTA	A	50 000 000
POSSIEDE	A	34 000 000
RICARICA	A	200 000 00
ACQUISTA	A	500 000 000
EVADE	A	490 000 000
PROPONE	A	80 000
APPARTIENE_R	A	4 857 000
APPARTIENE_P	A	5 000 000
SELEZIONA	A	2 500 000 000
CONSEGNA	A	490 000 000

Abbiamo stimato il numero di utenti a 34 milioni secondo i dati forniti dalla piattaforma di Delivery JustEat, di questi il 3% circa sono premium (1 milione). Il borsellino elettronico è univoco per ognuno degli utenti, mentre il numero di ricariche per utente è stato approssimato a 6. Considerando le 3 modalità di pagamento è stato considerato il numero di impostazioni di quest'ultime a poco meno di due per utente (circa 50 milioni). Abbiamo stimato un rider ogni 500 utenti e un ristorante ogni 700 utenti circa. Gli ordini totali sono stimati a 500 milioni mentre quelli evasi come ordini totali-ordini consegnati. Gli ordini cancellati sono stati calcolati come il 2% di quelli totali. Il numero di proposte è calcolato come ristoranti * piatti, considerando che in ogni locale un piatto può essere interpretato in modo differente (es: pizza margherita in pizzerie diverse). Le due associazioni Appartiene sono state ricavate moltiplicando il numero di ristoranti e piatti rispettivamente per categorie e liste. Nell'App Cibora, calcolando una media di 5 piatti per ordine, il numero di selezioni è due miliardi e mezzo. Per quanto riguarda la frequenza della chat abbiamo calcolato una media di 5 messaggi per ogni chat creata, divisa per due stimando che possano esserci delle chat vuote. Per quanto riguarda la frequenza degli invii e delle ricevute dei messaggi nelle chat, abbiamo stimato dei valori che sono uguali a due a due per le entità (stessi invii e ricevute per chat) e che sono stati calcolati moltiplicando la frequenza di quell'entità per la metà della frequenza degli ordini.

2.2 Tavola delle operazioni

DESCRIZIONE	TIPO	FREQUENZA
Registrazione nuovo utente	I	3 000 al giorno
Registrazione nuovo ristorante	I	50 al giorno
Inserimento consegna effettuata dal Rider	I	1 360 000 al giorno
Inserimento di nuovi messaggi	I	25 000 000 al giorno
Inserimento di nuovi ordini	I	5 000 000 al giorno
Aggiornamento dati del Rider, compreso il numero di consegne effettuate	B	2 720 000 al giorno
Ricarica Borsellino Elettronico	B	100 000 al giorno
Aggiornamento Modalità Premium	B	1 000 al giorno
Aggiornamento Stellette Ristorante	B	1 volta a settimana
Aggiornamento Classifiche *	B	1 volta al mese

Gestione Top Partner	B	20 al giorno
Ricerca della posizione del rider più vicino all'ordine *	B	5 000 000 al giorno
Stampa n consegne effettuate dal Rider	B	204 000 al giorno

Sono state inserite in tabella le operazioni ritenute più importanti, basandosi anche sui requisiti rivisti. La prima parte contiene un insieme di operazioni base di inserimento sulle principali entità coinvolte nello schema. Nella seconda parte le operazioni sono indicate come Batch (B) in quanto coinvolgono, tramite interrogazioni apposite, il DB. Sono, infine, state riportate con un * le due operazioni richieste esplicitamente dai requisiti del progetto, vedasi il testo e la parte delle operazioni per i dettagli.

I numeri sono stati stimati sulla base della popolazione dei dati esplicitata nella sezione 2.1. Il calcolo delle due operazioni presenti in sezione 2.3 sono riportati nelle tabelle della relativa sezione, mentre si segnala come la ricerca della posizione più vicina del rider per un ordine coincida, come frequenza, con il numero di ordini.

2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle Ridondanze

Elenco delle ridondanze riscontrate:

- Attributo “Ordini effettuati” sull’entità Utente
- Attributo “N consegne” sull’entità Rider
- Attributo “Prezzo tot” sull’entità Ordine

Analisi dell’attributo “N consegne”:

Tavola dei volumi:

CONCETTO	TIPO	VOLUME
ORDINE	E	500 000 000
CONSEGNA	A	490 000 000
RIDER	E	68 000

Tavola delle operazioni:

DESCRIZIONE	TIPO	FREQUENZA
Inserimento consegna effettuata dal Rider	I	1 360 000 al giorno (20 consegne * n rider)
Stampa n consegne effettuate dal Rider	B	204 000 al giorno (68 000 * colazione, pranzo, cena)

SCENARIO A:

Assenza di ridondanza

Operazione 1: Inserimento consegna effettuata dal Rider
Schema di operazione

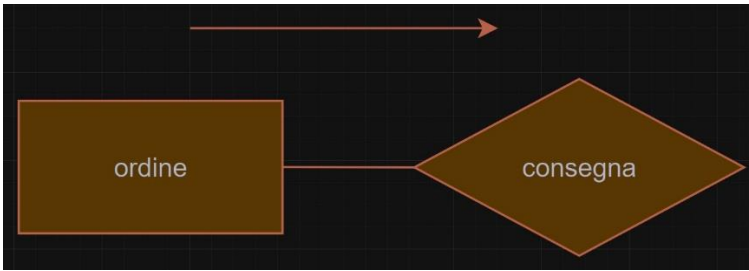
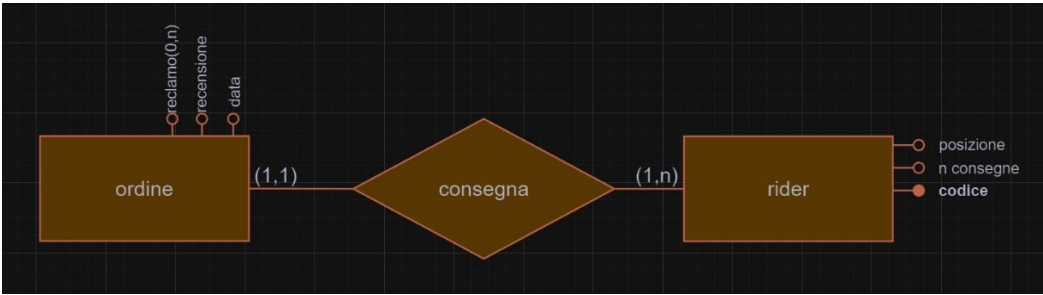


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
ORDINE	Entità	1	S
CONSEGNA	Associazione	1	S

SCENARIO A:
Assenza di ridondanza

Operazione 2: Stampa n consegne effettuate dal Rider
Schema di operazione

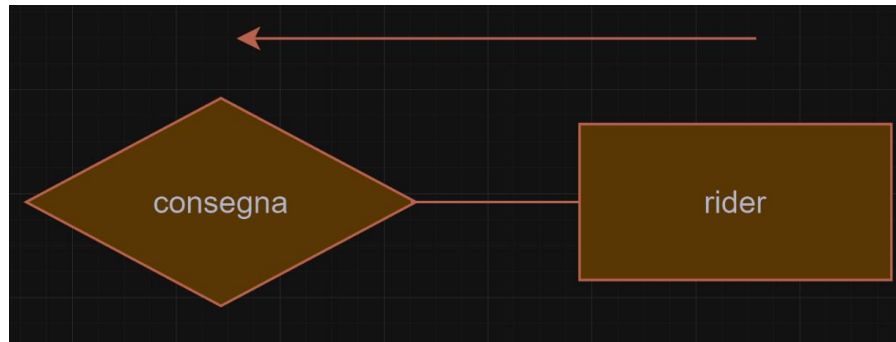


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
RIDER	Entità	1	L
CONSEGNA	Associazione	7 200 (n consegne/n rider)	L

SCENARIO B:
Presenza di ridondanza

Operazione 1: Inserimento consegna effettuata dal Rider
Schema di operazione

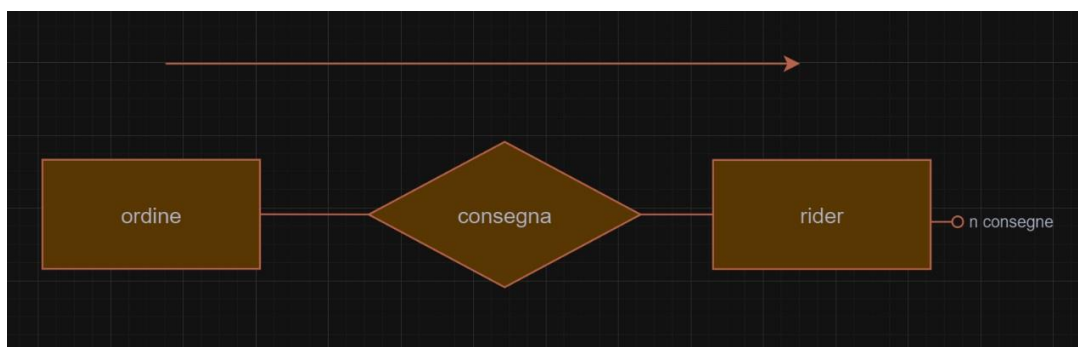


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
ORDINE	Entità	1	S
CONSEGNA	Associazione	1	S
RIDER	Entità	1	L
RIDER	Entità	1	S

SCENARIO B:

Presenza di ridondanza

Operazione 2: Stampa n consegne effettuate dal Rider
Schema di operazione

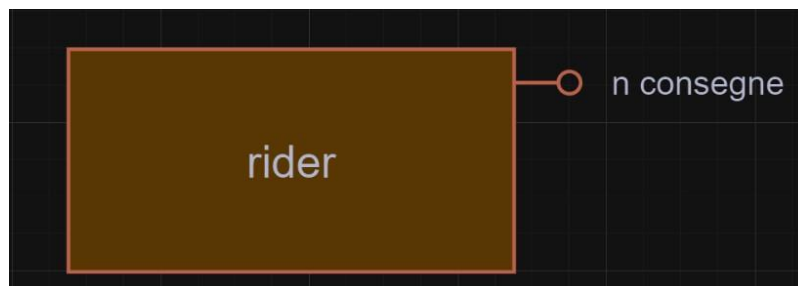


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
RIDER	Entità	1	L

ANALISI DEI COSTI E VALUTAZIONE FINALE:

Premessa: gli accessi in scrittura sono più onerosi quindi per fare una stima dei costi contiamo il costo di una scrittura come il doppio di una lettura.

SCENARIO A:

Assenza di ridondanza

- **Spazio:** 0 byte (si considera solo lo spazio occupato dalla ridondanza. Se non c'è l'attributo, lo spazio è 0)
- **Tempo:**
 1. **Operazione 1:** $2 * 1\,360\,000$ accessi in scrittura al giorno
 2. **Operazione 2:** $7201 * 204\,000$ accessi in lettura al giorno
 3. **Tempo totale giornaliero:** $(2 * 1\,360\,000) * 2 + (7\,201 * 204\,000) = 1\,474\,444\,000$ accessi totali al giorno

SCENARIO B:

Presenza di ridondanza

- **Spazio:** Assumiamo 1 byte per memorizzare il numero di consegne effettuate per ogni rider (256 consegne circa). $1 * 490\,000\,000$ (numero consegne)
- **Tempo:**
 4. **Operazione 1:** $3 * 1\,360\,000$ accessi in scrittura al giorno + $1 * 204\,000$ accessi in lettura al giorno
 5. **Operazione 2:** trascurabile (solo 1 accesso L)
 6. **Tempo totale giornaliero:** $(3 * 1\,360\,000) * 2 + (1 * 204\,000) = 8\,364\,000$ accessi totali al giorno

Nell'analizzare la presenza o meno di ridondanze ci sono certamente vari aspetti da tenere in considerazione e che possono anche determinare scelte diverse. Da una parte si può dare enfasi alle prestazioni, alla scalabilità e ai costi, dall'altra sono anche importanti la manutenzione/gestione e l'esperienza user. In definitiva, considerando il nostro obiettivo minimizzare i costi di storage, il primo scenario senza la ridondanza "n consegne" è preferibile.

2.3.2 Eliminazione delle generalizzazioni

Nello schema E-R sono state inserite le seguenti generalizzazioni: la Modalità di Pagamento può assumere i valori Satispay, Carta di Credito o Paypal (totale ed esclusiva). Una generalizzazione parziale è stata inserita sull'Utente per individuare anche il caso degli Utenti Premium (con una data di ingresso in questa sezione). Logica molto simile è stata utilizzata per intercettare gli ordini cancellati da Utenti o Ristoranti. Infine sono state introdotte due ulteriori generalizzazioni specifiche per i Rider.

2.3.3 Eventuale partizionamento/accorpamento di entità e associazioni

E' stato scelto già nella prima fase di ristrutturazione di scegliere meticolosamente gli attributi in modo che descrivessero solamente la singola entità, evitando possibili partizionamenti.

Discorso simile è stato fatto nella scelta del partizionamento per le associazioni. Focus di questo argomento potrebbero essere gli ordini, con le nostre scelte, tuttavia, è stato ritenuto superfluo considerando che gli ordini passati possono essere ricavati dagli attributi data/ora dell'associazione "Consegna". La gestione degli "Ordini Cancellati" avviene tramite una generalizzazione sugli "Ordini".

Infine, per l'accorpamento di entità è stato valutato di accorpare il borsellino elettronico nell'utente ma è abbiamo ritenuto importante separare le due cose per rappresentare meglio una porzione di realtà importante e distinta (borsellino).

2.3.4 Eventuale eliminazione degli attributi composti e degli attributi multivalore

Sono stati individuati, e verranno tradotti adeguatamente nel modello ristrutturato, i seguenti attributi multivalore: "Ingredienti" e "Allergeni" nell'entità "Piatti", "Codici Sconto" nell'entità "Borsellino Elettronico" e "Reclamo" su "Ordine".

Per quanto riguarda l'eliminazione degli attributi composti, nello schema E-R sono stati individuati i due attributi "Indirizzo" nelle entità "Utente" e "Ristorante".

2.3.5 Eventuale scelta degli identificatori principali

Sono state effettuate scelte piuttosto classiche nella decisione degli identificatori principali. In alcuni casi, come i nomi delle "Liste" di piatti e delle "Categorie" di ristoranti, sono stati individuati identificatori semplici ed univoci. Altre volte è stato fatto uso di identificatori esterni, come l'id del "Borsellino Elettronico" che si identifica tramite l'Utente. Nel caso specifico dei reclami, è stato rimosso l'identificatore esterno sugli ordini in quanto identificabile semplicemente dall'ID incrementale e avrebbe creato problemi nella fase di progettazione dello schema relazionale (due attributi uguali nell'associazione). In altri casi, infine, sono stati usati identificatori incrementali per semplificare la gestione, in particolare per le Chat e gli Ordini. Inoltre, in entrambi i casi, tramite le associazioni binarie con le altre entità è piuttosto semplice ricavare i collegamenti (es: a quale Ordine si riferisce una Chat tramite apposita associazione).

[illegible]

Integrità:

- 1- Borsellino Elettronico.Saldo aggiornato ad ogni ordine (Saldo = Saldo-Ordine.PrezzoTot).
- 2-Per poter effettuare un ordine, Borsellino Elettronico.Saldo >= Ordine.PrezzoTot.
- 3-La chat viene creata tra solamente due entità contemporaneamente (utente-ristorante, utente-rider,ristorante-rider).
- 4- Ristorante.nStellette=somma voti Ordine.Stellette (riferite a quel ristorante)/n ordini con valutazione, aggiornato ogni settimana.
- 5-I ristoranti con queste caratteristiche sono Top Partner: almeno 20 ordini consegnati correttamente, “n stellette” >= 4.5, una percentuale massima di ordini annullati dal ristorante dell’1.5%, una percentuale massima di ordini con reclami del 2.5%), viene salvata la data nell’entità Utente.
- 6-Il pagamento dell'ordine consente all'utente di:
effettuare reclami, chattare con il ristorante e con il rider, inserire una recensione. Per effettuare ordine Acquista.Pagato=TRUE.
- 7-Al momento dell'ordine, il sistema Cibora 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.
- 8-Ci sono due tipi di sconti:
il primo è quello che il ristoratore imposta sul piatto (Piatto.Sconto),
il secondo tipo di sconto (Borsellino.CodiciSconto), invece, viene ottenuto dal cliente in base al numero e all'importo degli ordini passati.
-dopo il quinto ordine >> 20 per cento

-se l'utente spende più di 20 euro, riceve uno sconto del 15%

-se l'utente effettua almeno un ordine a settimana per 4 settimane, ottiene uno sconto del 50%.

Sarà cura dell'applicativo mostrare i codici sconto posseduti dall'utente al momento dell'ordine.

9-Se il piatto appartiene ad una determinata lista y, e il ristorante propone quel piatto, allora il ristorante entrerà a far parte di una categoria x legata alla lista y.

10-Ristorante.DataTopPartner se è diverso da NULL, significa che il Ristorante fa parte della categoria Top Partner.

11-Ordine.Commento e Ordine.NStellette possono essere valorizzati se l'Utente ha fatto la recensione.

12-Rider.Autonomia è diverso da NULL solamente nel caso in cui il rider possiede mezzo elettrico (Km nella versione precedente).

13-Rider.Stato e Rider.MezzoTrasporto sono i dati che il Rider imposta in tempo reale, devono assumere sempre un solo valore dei 3 possibili (vedi generalizzazione modello non ristrutturato).

Derivazione:

- Rider.nConsegne è derivabile dalla partecipazione di Rider in Consegna
- Ordine.PrezzoTot è derivabile dalla somma di ogni Piatto.Prezzo nell'Ordine

2.5 Schema Relazionale

Utente(Email, Nome, Password, Telefono, CAP, Via, Numero, OrdiniEffettuati, UtentePremium*)

Modalità di Pagamento(Nome, Utente)

Borsellino Elettronico(Utente, Saldo)

Ricarica(Utente, Quantità)

Codice Sconto(Tipo, Valore)

Conserva(Borsellino Elettronico, Codice Sconto)

Ristorante(Nome, Descrizione, Costo Spedizione, Img Profilo, N Stellette, CAP, Via, Numero, Data Top Partner*)

Categoria(Nome)

Appartiene_r(Categoria, Ristorante)

Piatto(Titolo, Ristorante, Sconto*, Prezzo, Img)

Propone(Ristorante, Piatto)

Lista(Nome)

Appartiene_p(Piatto, Lista)

Allergeni(Id, Nome)

Ingredienti(Id, Nome)

Composto_A(Allergeni Piatto, Titolo, Ristorante)

Composto_I(Ingredienti Piatto, Titolo, Ristorante)

Ordine(Id, Cancellato*, Prezzo Tot, Commento, Stellette, Data, Ora)

Acquista(Ordine, Utente, Pagato*)

Evade(Ordine, Ristorante)

Seleziona(Ordine, Piatto, Ristorante)

Reclami(Titolo, Msg)

Possiede_r(Reclamo, Ordine)

Rider(Codice, Posizione, Autonomia*, Stato, Mezzo di Trasporto)

Consegna(Ordine, Rider, Ora Consegna Rider, Ora Consegna Utente)

Chat(Id, Messaggio, Ora)

Genera(Ordine, Chat)

Invia_u(Utente, Chat)

Riceve_u(Utente, Chat)

Invia_ri(Ristorante, Chat)

Riceve_ri(Ristorante, Chat)

Invia_ra(Rider, Chat)

Riceve_ra(Rider, Chat)

Vincoli:

Unique(Password, Telefono) in **Utente**

Modalità di Pagamento(Utente) referencia **Utente**(Email)

Borsellino Elettronico(Utente) referencia **Utente**(Email)

Ricarica(Utente) referencia **Borsellino Elettronico**(Utente)

Conserva(Borsellino Elettronico) referencia **Borsellino Elettronico**(Utente)

Conserva(Codice Sconto) referencia **Codice Sconto**(Tipo)

Appartiene_r(Categoria) referencia **Categoria**(Nome)

Appartiene_r(Ristorante) referencia **Ristorante**(Nome)

Piatto(Ristorante) referencia **Ristorante**(Nome)

Propone(Ristorante) referencia **Ristorante**(Nome)

Propone(Piatto) referencia **Piatto**(Titolo)

Appartiene_p(Lista) referencia **Lista**(Nome)

Appartiene_p(Piatto) referencia **Piatto**(Titolo)

Composto_a(Allergeni Piatto) referencia **Allergeni**(Id)

Composto_a(Piatto, Ristorante) referencia **Piatto**(Titolo, Ristorante)

Composto_i(Ingredienti Piatto) referencia **Ingredienti**(Id)

Composto_i(Piatto, Ristorante) referencia **Piatto**(Titolo, Ristorante)

Acquista(Utente) referencia **Utente**(Email)
Acquista(Ordine) referencia **Ordine**(Id)
Evade(Ristorante) referencia **Ristorante**(Nome)
Evade(Ordine) referencia **Ordine**(Id)
Seleziona(Piatto, Ristorante) referencia **Piatto**(Titolo, Ristorante)
Seleziona(Ordine) referencia **Ordine**(Id)
Possiede_r(Reclamo) referencia **Reclami**(Titolo)
Possiede_r(Ordine) referencia **Ordini**(Id)
Consegna(Ordine) referencia **Ordini**(Id)
Consegna(Rider) referencia **Rider**(Codice)
Genera(Ordine) referencia **Ordini**(Id)
Genera(Chat) referencia **Chat**(Id)
Invia_u(Utente) referencia **Utente**(Email)
Invia_u(Chat) referencia **Chat**(Id)
Riceve_u(Utente) referencia **Utente**(Email)
Riceve_u(Chat) referencia **Chat**(Id)
Invia_ri(Ristorante) referencia **Ristorante**(Nome)
Invia_ri(Chat) referencia **Chat**(Id)
Riceve_ri(Ristorante) referencia **Ristorante**(Nome)
Riceve_ri(Chat) referencia **Chat**(Id)
Invia_ra(Rider) referencia **Rider**(Codice)
Invia_ra(Chat) referencia **Chat**(Id)
Riceve_ra(Rider) referencia **Rider**(Codice)
Riceve_ra(Chat) referencia **Chat**(Id)

3.0 IMPLEMENTAZIONE

3.1 DDL di creazione del Database

```

CREATE TABLE Utente (
  Email VARCHAR(255) PRIMARY KEY,
  Nome VARCHAR(255) NOT NULL,
  Password VARCHAR(255) NOT NULL,
  Telefono VARCHAR(20) NOT NULL,
  CAP VARCHAR(10),
  Via VARCHAR(255),
  Numero INT,
  OrdiniEffettuati INT,

```

```

    UtentePremium BOOLEAN,
    UNIQUE (Password, Telefono)
);
CREATE TABLE ModalitaDiPagamento (
    Nome VARCHAR(255),
    Utente VARCHAR(255),
    PRIMARY KEY (Nome, Utente),
    FOREIGN KEY (Utente) REFERENCES Utente(Email)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE BorsellinoElettronico (
    Utente VARCHAR(255) PRIMARY KEY,
    Saldo DECIMAL(10,2),
    FOREIGN KEY (Utente) REFERENCES Utente(Email)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE Ricarica (
    Utente VARCHAR(255),
    Quantita DECIMAL(10,2),
    FOREIGN KEY (Utente) REFERENCES BorsellinoElettronico(Utente)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE CodiceSconto (
    Tipo VARCHAR(255) PRIMARY KEY,
    Valore DECIMAL(10,2)
);
CREATE TABLE Conserva (
    BorsellinoElettronico VARCHAR(255),
    CodiceSconto VARCHAR(255),
    PRIMARY KEY (BorsellinoElettronico, CodiceSconto),
    FOREIGN KEY (BorsellinoElettronico) REFERENCES BorsellinoElettronico(Utente)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY (CodiceSconto) REFERENCES CodiceSconto(Tipo)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE Ristorante (
    Nome VARCHAR(255) PRIMARY KEY,
    Descrizione TEXT,
    CostoSpedizione DECIMAL(10,2),
    ImgProfilo VARCHAR(255),
    NStellette INT,
    CAP VARCHAR(10),
    Via VARCHAR(255),
    Numero INT,
    DataTopPartner DATE
);
CREATE TABLE Categoria (
    Nome VARCHAR(255) PRIMARY KEY
);
CREATE TABLE Appartiene_r (
    Categoria VARCHAR(255),

```



```

Ristorante VARCHAR(255),
PRIMARY KEY (Categoria, Ristorante),
FOREIGN KEY (Categoria) REFERENCES Categoria(Nome)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)
    ON UPDATE CASCADE
    ON DELETE CASCADE

```

```
);
```

```

CREATE TABLE Piatto (
    Titolo VARCHAR(255),
    Ristorante VARCHAR(255),
    Sconto VARCHAR(255),
    Prezzo DECIMAL(10,2),
    Img VARCHAR(255),
    PRIMARY KEY (Titolo, Ristorante),
    FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE

```

```
);
```

```

CREATE TABLE Propone (
    Ristorante VARCHAR(255),
    Piatto VARCHAR(255),
    FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Piatto) REFERENCES Piatto(Titolo)
        ON UPDATE CASCADE
        ON DELETE CASCADE

```

```
);
```

```

CREATE TABLE Propone (
    Ristorante VARCHAR(255),
    Piatto VARCHAR(255),
    FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Piatto) REFERENCES Piatto(Titolo)
        ON UPDATE CASCADE
        ON DELETE CASCADE

```

```
);
```

```

CREATE TABLE Lista (
    Nome VARCHAR(255) PRIMARY KEY

```

```
);
```

```

CREATE TABLE Appartiene_p (
    Piatto VARCHAR(255),
    Lista VARCHAR(255),

```

```

PRIMARY KEY (Piatto, Lista),
FOREIGN KEY (Piatto) REFERENCES Piatto(Titolo)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Lista) REFERENCES Lista(Nome)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE Allergeni (
    Id INT PRIMARY KEY,
    Nome VARCHAR(255)
);
CREATE TABLE Ingredienti (
    Id INT PRIMARY KEY,
    Nome VARCHAR(255)
);
CREATE TABLE Composto_A (
    AllergeniPiatto INT,
    Titolo VARCHAR(255),
    Ristorante VARCHAR(255),
    FOREIGN KEY (AllergeniPiatto) REFERENCES Allergeni(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Titolo, Ristorante) REFERENCES Piatto(Titolo, Ristorante)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
CREATE TABLE Composto_I (
    IngredientiPiatto INT,
    Titolo VARCHAR(255),
    Ristorante VARCHAR(255),
    FOREIGN KEY (IngredientiPiatto) REFERENCES Ingredienti(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,

```

```

FOREIGN KEY (Titolo, Ristorante) REFERENCES Piatto(Titolo, Ristorante)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE Ordine (
    Id INT PRIMARY KEY,
    Cancellato BOOLEAN,
    PrezzoTot DECIMAL(10,2),
    Commento TEXT,
    Stellette INT,
    Data DATE,
    Ora TIME
);
CREATE TABLE Acquista (
    Ordine INT,
    Utente VARCHAR(255),
    Pagato BOOLEAN,
    FOREIGN KEY (Ordine) REFERENCES Ordine(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Utente) REFERENCES Utente(Email)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
CREATE TABLE Evade (
    Ordine INT,
    Ristorante VARCHAR(255),
    FOREIGN KEY (Ordine) REFERENCES Ordine(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

```
CREATE TABLE Seleziona (  
    Ordine INT,  
    Piatto VARCHAR(255),  
    Ristorante VARCHAR(255),  
    FOREIGN KEY (Ordine) REFERENCES Ordine(Id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (Piatto, Ristorante) REFERENCES Piatto(Titolo, Ristorante)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE Reclami (  
    Titolo VARCHAR(255) PRIMARY KEY,  
    Msg TEXT  
);
```

```
CREATE TABLE Possiede_r (  
    Reclamo VARCHAR(255),  
    Ordine INT,  
    FOREIGN KEY (Reclamo) REFERENCES Reclami(Titolo)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (Ordine) REFERENCES Ordine(Id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE Rider (  
    Codice VARCHAR(255) PRIMARY KEY,  
    Posizione VARCHAR(255),  
    Autonomia BOOLEAN,  
    Stato VARCHAR(255),  
    MezzoDiTrasporto VARCHAR(255)  
);
```

```
CREATE TABLE Consegna (  
    Ordine INT,
```

```

Rider VARCHAR(255),
OraConsegnaRider TIME,
OraConsegnaUtente TIME,
FOREIGN KEY (Ordine) REFERENCES Ordine(Id)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Rider) REFERENCES Rider(Codice)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE Chat (
    Id INT PRIMARY KEY,
    Messaggio TEXT,
    Ora TIME
);
CREATE TABLE Genera (
    Ordine INT,
    Chat INT,
    FOREIGN KEY (Ordine) REFERENCES Ordine(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Chat) REFERENCES Chat(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
CREATE TABLE Invia_u (
    Utente VARCHAR(255),
    Chat INT,
    FOREIGN KEY (Utente) REFERENCES Utente(Email)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Chat) REFERENCES Chat(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE

```

```
);  
CREATE TABLE Riceve_u (  
    Utente VARCHAR(255),  
    Chat INT,  
    FOREIGN KEY (Utente) REFERENCES Utente(Email)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (Chat) REFERENCES Chat(Id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```
);  
CREATE TABLE Invia_ri (  
    Ristorante VARCHAR(255),  
    Chat INT,  
    FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (Chat) REFERENCES Chat(Id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```
);  
CREATE TABLE Riceve_ri (  
    Ristorante VARCHAR(255),  
    Chat INT,  
    FOREIGN KEY (Ristorante) REFERENCES Ristorante(Nome)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (Chat) REFERENCES Chat(Id)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```
);  
CREATE TABLE Invia_ra (  
    Rider VARCHAR(255),  
    Chat INT,
```

```

FOREIGN KEY (Rider) REFERENCES Rider(Codice)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (Chat) REFERENCES Chat(Id)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);
CREATE TABLE Riceve_ra (
    Rider VARCHAR(255),
    Chat INT,
    FOREIGN KEY (Rider) REFERENCES Rider(Codice)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (Chat) REFERENCES Chat(Id)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

3.2 DML di popolamento di tutte le tabelle del Database

```

INSERT INTO Utente VALUES ('laura.martini@gmail.com', 'Laura Martini', 'laumarti02!', '3201234567',
'00100', 'Via Roma', 12, 5, TRUE);

```

```

INSERT INTO Utente VALUES ('franco.verdi@gmail.com', 'Franco Verdi', 'franco789!', '3207654321',
'00122', 'Via Firenze', 23, 2, FALSE);

```

```

INSERT INTO ModalitaDiPagamento VALUES ('Carta di Credito', 'laura.martini@gmail.com');

```

```

INSERT INTO ModalitaDiPagamento VALUES ('PayPal', 'franco.verdi@gmail.com');

```

```

INSERT INTO BorsellinoElettronico VALUES ('laura.martini@gmail.com', 250.00);

```

```

INSERT INTO BorsellinoElettronico VALUES ('franco.verdi@gmail.com', 75.00);

```

```

INSERT INTO Ricarica VALUES ('laura.martini@gmail.com', 100.00);

```

```

INSERT INTO Ricarica VALUES ('franco.verdi@gmail.com', 50.00);

```

```

INSERT INTO CodiceSconto VALUES ('PRIMAVERA20', 20);

```

INSERT INTO CodiceSconto VALUES ('ESTATE50', 50);

INSERT INTO Conserva VALUES ('laura.martini@gmail.com', 'PRIMAVERA20');

INSERT INTO Conserva VALUES ('franco.verdi@gmail.com', 'ESTATE50');

INSERT INTO Ristorante VALUES ('Il Vecchio Forno', 'Ristorante tradizionale italiano', 3.50, 'forno.jpg', 5, '00100', 'Viale Europa', 5, '2019-05-15');

INSERT INTO Ristorante VALUES ('Tokyo Sushi', 'Cucina giapponese contemporanea', 2.00, 'sushi.jpg', 4, '00122', 'Corso Asia', 8, NULL);

INSERT INTO Categoria VALUES ('Cucina Italiana');

INSERT INTO Categoria VALUES ('Cucina Giapponese');

INSERT INTO Appartiene_r VALUES ('Cucina Italiana', 'Il Vecchio Forno');

INSERT INTO Appartiene_r VALUES ('Cucina Giapponese', 'Tokyo Sushi');

INSERT INTO Piatto VALUES ('Tagliatelle al Ragu', 'Il Vecchio Forno', NULL, 12.50, 'tagliatelle.jpg');

INSERT INTO Piatto VALUES ('Nigiri Sushi', 'Tokyo Sushi', 'ESTATE50', 18.00, 'nigiri.jpg');

INSERT INTO Propone VALUES ('Il Vecchio Forno', 'Tagliatelle al Ragu');

INSERT INTO Propone VALUES ('Tokyo Sushi', 'Nigiri Sushi');

INSERT INTO Lista VALUES ('Pranzo di Lavoro');

INSERT INTO Lista VALUES ('Cena Romantica');

INSERT INTO Appartiene_p VALUES ('Tagliatelle al Ragu', 'Pranzo di Lavoro');

INSERT INTO Appartiene_p VALUES ('Nigiri Sushi', 'Cena Romantica');

INSERT INTO Allergeni VALUES (1, 'Glutine');

INSERT INTO Allergeni VALUES (2, 'Pesce');

INSERT INTO Ingredienti VALUES (1, 'Farina');

INSERT INTO Ingredienti VALUES (2, 'Tonno');

INSERT INTO Composto_A VALUES (1, 'Tagliatelle al Ragu', 'Il Vecchio Forno');

INSERT INTO Composto_A VALUES (2, 'Nigiri Sushi', 'Tokyo Sushi');

INSERT INTO Composto_I VALUES (1, 'Tagliatelle al Ragu', 'Il Vecchio Forno');

INSERT INTO Composto_I VALUES (2, 'Nigiri Sushi', 'Tokyo Sushi');

INSERT INTO Ordine VALUES (1, FALSE, 25.00, 'Molto soddisfatto, delizioso!', 5, '2023-04-15', '13:20');

INSERT INTO Ordine VALUES (2, TRUE, 40.00, 'Purtroppo era freddo al momento dell'arrivo.', 2, '2023-04-16', '20:45');

INSERT INTO Acquista VALUES (1, 'laura.martini@gmail.com', TRUE);

INSERT INTO Acquista VALUES (2, 'franco.verdi@gmail.com', FALSE);

INSERT INTO Evade VALUES (1, 'Il Vecchio Forno');

INSERT INTO Evade VALUES (2, 'Tokyo Sushi');

INSERT INTO Seleziona VALUES (1, 'Tagliatelle al Ragu', 'Il Vecchio Forno');

INSERT INTO Seleziona VALUES (2, 'Nigiri Sushi', 'Tokyo Sushi');

INSERT INTO Reclami VALUES ('ReclamoTempo', 'Tempo di attesa troppo lungo');

INSERT INTO Reclami VALUES ('ReclamoQualita', 'Qualità del cibo non soddisfacente');

INSERT INTO Possiede_r VALUES ('ReclamoTempo', 1);

INSERT INTO Possiede_r VALUES ('ReclamoQualita', 2);

INSERT INTO Rider VALUES ('R12345', 'Zona Centro', TRUE, 'Disponibile', 'Bicicletta');

INSERT INTO Rider VALUES ('R67890', 'Zona Nord', FALSE, 'In consegna', 'Monopattino');

INSERT INTO Consegna VALUES (1, 'R12345', '14:00', '14:30');

INSERT INTO Consegna VALUES (2, 'R67890', '21:00', '21:35');

INSERT INTO Chat VALUES (1, 'Quando arriverà il mio ordine?', '13:05');

INSERT INTO Chat VALUES (2, 'Grazie per la vostra gentilezza e rapidità!', '21:10');

INSERT INTO Genera VALUES (1, 1);

INSERT INTO Genera VALUES (2, 2);

```
INSERT INTO Invia_u VALUES ('laura.martini@gmail.com', 1);
```

```
INSERT INTO Invia_u VALUES ('franco.verdi@gmail.com', 2);
```

```
INSERT INTO Riceve_u VALUES ('laura.martini@gmail.com', 1);
```

```
INSERT INTO Riceve_u VALUES ('franco.verdi@gmail.com', 2);
```

```
INSERT INTO Invia_ri VALUES ('Il Vecchio Forno', 1);
```

```
INSERT INTO Invia_ri VALUES ('Tokyo Sushi', 2);
```

```
INSERT INTO Riceve_ri VALUES ('Il Vecchio Forno', 1);
```

```
INSERT INTO Riceve_ri VALUES ('Tokyo Sushi', 2);
```

```
INSERT INTO Invia_ra VALUES ('R12345', 1);
```

```
INSERT INTO Invia_ra VALUES ('R67890', 2);
```

```
INSERT INTO Riceve_ra VALUES ('R12345', 1);
```

```
INSERT INTO Riceve_ra VALUES ('R67890', 2);
```

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

```
DELETE FROM Ristorante WHERE Nome = 'Il Vecchio Forno';
```

```
UPDATE Utente SET Email = 'laura.nuova@gmail.com' WHERE Email = 'laura.martini@gmail.com';
```

```
UPDATE CodiceSconto SET Tipo = 'ESTATE60' WHERE Tipo = 'ESTATE50';
```

```
ALTER TABLE Utente
```

```
ADD DataNascita DATE;
```

```
SELECT
```

```
    P.Titolo,
```

```
    P.Ristorante
```

```
FROM
```

Piatto P

JOIN

Composto_I CI ON P.Titolo = CI.Titolo AND P.Ristorante = CI.Ristorante

JOIN

Ingredienti I ON CI.IngredientiPiatto = I.Id

WHERE

I.Nome IN ('salmone', 'grana padano')