

Progetto Basi di Dati

Sandri Gabriele 1041535
Sandri Mattia 1041531

Laboratorio Basi di Dati 2023/2024

Progetto di piattaforma di food delivery

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

Sommario

.....	1
Progettazione Concettuale:	6
1. Glossario dei termini:	6
2. Strutturazione dei requisiti rivisti in gruppi di frasi omogenee:	7
1. Generale:	7
2. User.....	7
3. Utente:.....	7
4. Ristorante:	7
5. Rider:	7
6. Piatto:	8
7. Ordine:.....	8
8. Valutazione:.....	8
9. Chat:	8
10. Metodo di pagamento:.....	8
11. Categoria:	8
12. Sconto	8
3. Schema E-R:	9
4. Regole Aziendali:.....	10
1. Dizionario Entità:.....	10
2. Dizionario Associazioni:	11
3. Vincoli di Integrità:.....	12
4. Derivazioni:	12
Progettazione Logica:	13
1. Tavola dei Volumi:	13
2. Operazioni più significative:	14
3. Analisi delle Ridondanze:	15
1. Analisi dettagliata: Voto Ristorante:	15
4. Eliminazione delle Generalizzazioni:.....	18
Soluzioni ipotizzate per l'eliminazione:	18
5. Schema E-R Ristrutturato:	19
6. Commento:	20
1. Commento schema ER Ristrutturato:	20
2. Accorpamento di Entità o Relazioni:.....	20
3. Scelta degli identificatori principali:.....	20
7. Regole Aziendali: Vincoli di Integrità:	20
Traduzione Concettuale - Logica:	21
1. Schema Logico:	21
2. Vincoli di integrità referenziale:	22
DDL:	23
1. Commento DDL:	26
DML:	27
Modifiche e Cancellazioni per la verifica dei vincoli:	28

Progettazione Concettuale:

1. Glossario dei termini:

TERMINI	DESCRIZIONE	SININIMI	COLLEGAMENTI
Utente	Persona che effettua gli ordini	Cliente	Ordini, Borsellino, Valutazione, Reclami, Utente
Ristorante	Attività che prepara gli Ordini	-	Valutazione, Ordini, Piatti, Consegnna
Rider	Persona che effettua le consegne	Fattorino, Ciclo-Fattorino	Valutazione, Ordine, Consegnna
Piatto	Elemento dell'ordine che viene preparato dal ristorante	Portata, Cibo, Pietanza	Ristorante, Ordine
Ordine	Insieme di Piatti	Cibo da consegnare Ordinazione	Ristorante, Utente, Consegnna, Valutazione, Piatto, Reclami
Valutazione	Recensione di un Ristorante o Rider	-	Rider, Ristorante, Utente, Ordine, Top P.
Reclamo	Incorrettezze in un ordine	-	Utente, Ristorante, Ordine
Consegnna	Ordine che viene consegnato dal rider	-	Rider, Ordine
Metodo di Pagamento	Mezzo attraverso il quale l'utente ricarica il borsellino.	Carta di credito, Bancomat	Utente
Sconto	Sconto applicato all'ordine in base al numero di essi effettuato	Codice Sconto	Ordine
Chat	Metodo di comunicazione fra Utente, Ristorante	-	Utente, Rider, Ristorante, Ordine
Borsellino	Conto Virtuale dell'utente, utilizzato per pagare gli ordini	-	Utente, Metodo di pag.
Top Partner	Speciale attestato ai migliori ristoranti	Speciale badge	Ristoranti, Consegnna, Valutazione, Ordini
Premium	Attributo utente che garantisce priorità sugli ordini	Modalità Premium	Utente, Ordine
Ordini Annullati	Ordini annullati	-	Ristoranti, Utente, Ordine
Categoria	Categoria piatti e ristorante	-	Piatto

2. Strutturazione dei requisiti rivisti in gruppi di frasi omogenee:

NOTA: Ai fini della progettazione è stato introdotto un nuovo requisito USER, per semplificare la progettazione concettuale;

1. Generale:

Si deve progettare la base di dati per “Cibora”, un innovativo servizio di food-delivery, per gestire i dati: dei ristoranti, degli utenti con i loro relativi ordini, e dei rider che effettuano le consegne.

2. User

Per User identificato da Identificativo rappresentiamo identificativo e tipo;

-Tipo: attributo che specifica il tipo di user: “Rider”, “Ristorante”, “Utente”

-Identificativo: stringa che varia in base al tipo (Codice Rider / Nome Ristorante / E-mail Utente);

3. Utente:

Per Utente identificato da identificativo rappresentiamo: nome, password, numero di telefono, indirizzo, borsellino, premium.

-Borsellino elettronico: cifra numerica equivalente al saldo disponibile dell’utente, ricaricabile attraverso metodo di pagamento;

-Identificativo -> E-mail: identificativo univoco dell’utente;

4. Ristorante:

Per Ristorante identificato da identificativo rappresentiamo: descrizione, indirizzo, costo della spedizione, immagine, voto, top partner, n° ordini annullati dal ristorante, n° ordini con reclamo, time-stamp.

-Identificativo -> Nome: identificativo univoco del ristorante;

-Voto: numero da 1 a 5, aggiornato ogni lunedì sulla base della percentuale di recensioni positive dell’ultima settimana;

-Descrizione: commento testuale di cosa vende il ristorante;

-Top Partner: time-stamp di entrata, che attesta il servizio eccellente del ristorante, dedicato ai ristoranti con almeno 20 ordini consegnanti correttamente e voto di almeno 4.5, percentuale massima di ordini annullati del 1.5%, percentuale massima di ordini con reclami di 2.5%; (NULL se non fa parte della categoria);

5. Rider:

Per Rider identificato da identificativo rappresentiamo: stato, mezzo, posizione, km rimanenti (solo per rider in monopattino), voto.

Identificativo -> Codice: identificativo univoco;

-Stato: Valore a stringa di “occupato”, “fuori servizio”, “disponibile”;

-Posizione: posizione aggiornata in tempo reale del rider, sotto forma di stringa di numeri;

-km Rimani: km rimanenti prima che si scarichi la batteria dei rider che come mezzo hanno un monopattino.

6.Piatto:

Per Piatto, identificato da titolo, identificativo ristorante, rappresentiamo: immagine, lista ingredienti, lista allergeni, prezzo, sconto, categoria.

-Titolo: nome/spiegazione del piatto;

-Lista ingredienti: stringa contenente tutti gli ingredienti separati da virgola.

-Lista allergeni: stringa contenente tutti gli allergeni separati da virgola.

-Categoria: un piatto appartiene a una o più liste (es. "i più venduti, promozioni, dolce...").

7.Ordine:

Per Ordine identificato da identificativo utente e time-stamp, rappresentiamo: stato, identificativo rider.

(Quando l'ordine viene consegnato si aggiungono gli attributi time-stamp consegna e mancia);
(L'ordine può essere annullato solo in fase di preparazione);

-Stato: valore testuale riguardante lo stato dell'ordine, che permette l'annullamento prima dell'affidamento al rider;

-Identificativo rider: codice del rider che effettua la consegna.

-Mancia: valore numerico;

8.Valutazione:

Per Valutazione su ordini passati rappresentiamo: identificativo utente, identificativo rider/identificativo ristorante, voto, commento testuale, time-stamp ordine.

-Voto: numero da 1 a 5;

9. Chat:

Per Chat identificata destinatario, mittente, time-stamp rappresentiamo: messaggio testuale, time-stamp ordine, identificativo utente, tipo;

-Tipo: è un attributo testuale che identifica il tipo della chat se di "reclamo" o "info";

10.Metodo di pagamento:

Per Metodo di Pagamento rappresentiamo: identificativo utente, intestatario, codice carta, codice segreto, scadenza carta.

11.Categoria:

Per Categoria, identificata da nome, rappresentiamo nome, titolo piatto, identificativo ristorante;

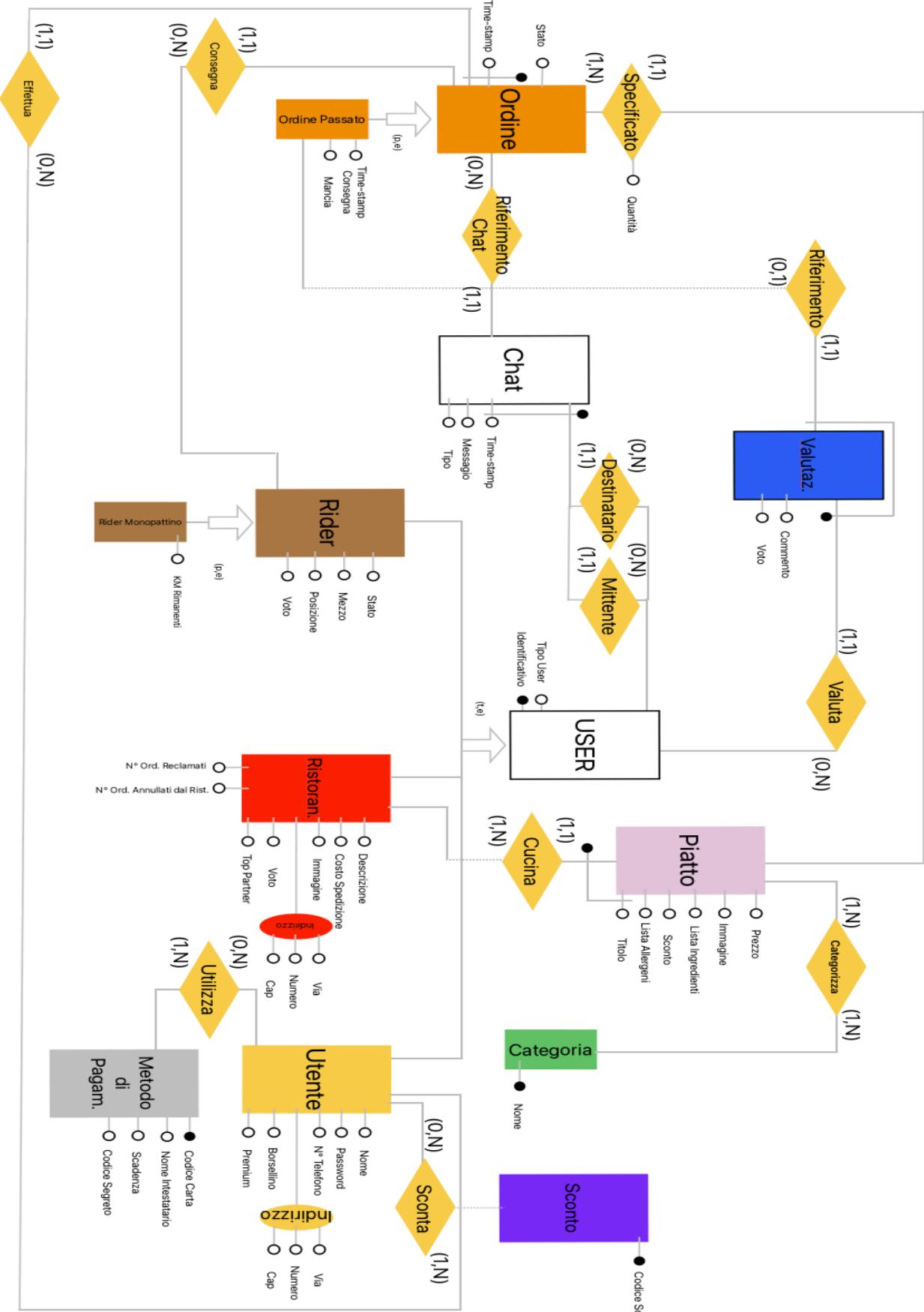
12. Sconto

Per Sconto identificato da codice sconto, rappresentiamo codice sconto, e il valore percentuale.

13. Dati appresi dalle schermate

Quando l'utente visualizza la lista dei ristoranti, sulla schermata dell'applicazione, oltre a tutte le informazioni dei ristornati, deve poter vedere il numero di valutazioni per ciascuno.

3. Schema E-R:



4. Regole Aziendali:

1. Dizionario Entità:

Entità	Descrizione	Attributi	Identificatore
Utente	Persona che effettua gli ordini, specializzato da User (T,E)	Nome, Password, N° telefono, Indirizzo, Borsellino, Premium	Identificativo
Ristoranti	Attività che prepara gli Ordini, specializzato da User (T,E)	Descrizione, Costo spedizione, Immagine, Indirizzo, Voto, Top Partner, N°ordini ann., N°ordini rec.	Identificativo
Rider	Persona che effettua le consegne, specializzato da User (T,E)	Stato, Mezzo, Posizione, Voto	Identificativo
Rider Monopattino	Generalizzazione di Rider (P,E)	Codice, Stato, Mezzo, Posizione, Voto, km rimanenti	Identificativo
User	Utilizzatore dell'applicativo	Tipo, Identificativo	Identificativo
Ordine	Insieme di dettagli ordine	Time-stamp, Stato	Time-stamp, Identificativo utente
Ordine Passato	Generalizzazione di Ordine, per gli ordini passati (P,E)	Time-stamp , Time-stamp Consegnata, Mancia, Stato	Time-stamp, Identificativo utente
Chat	Messaggio tra destinatario e mittente	Time-stamp , Messaggio, Tipo	Destinatario, Mittente, Time-stamp
Valutazione	Valutazione di un Ristorante o Rider	Commento e Voto	Ident. utente, Iden. rist. / Iden.rid. Time-stamp ordine
Metodo di pagamento	Mezzo attraverso il quale l'utente ricarica il borsellino.	Codice carta, Nome intestatario, Scadenza, codice segreto	Codice carta
Categoria	Categoria piatti e ristorante	Nome	Nome
Sconto	Sconto applicabile a un utente per i suoi ordini	Codice Sconto, Valore	Codice Sconto
Piatto	Piatto cucinato da Rist.	Prezzo, Immagine, List. Ingredienti, Sconto, List. Allerg., Titolo	Titolo, Iden. Rist.

2. Dizionario Associazioni:

Associazione	Descrizione	Componenti	Attributi
Utilizza	Metodo di pagamento di un utente	Utente (0, N), Metodo Pag. (1, N)	-
Effettua	Utente effettua ordine	Utente (0, N) Ordine (1, 1)	-
Scrive	User scrive valutazione	User (0, N), Valutazione (1,1)	-
Cucina	Ristorante cucina piatto	Ristorante (1, N), Piatto (1, 1)	-
Valuta	Valutazione valuta User	Valutazione (1,1), User(0,N)	-
Consegna	Rider consegna ordine	Rider (0, N) Ordine(1, 1)	-
Specificato	Ordine specificato da piatto	Piatto (0, N) Ordine(1, N)	Quantità
Categorizza	Categoria categorizza piatto	Piatto (1, N) Categoria(1, N)	-
Riferimento	Valutazione fa riferimento a ordine	Ordine (0, 1) Valutazione(1, 1)	-
Riferimento Chat	Chat si riferisce a un ordine/ordine pass.	Ordine (0, N) Chat(1, 1)	-
Destinatario	User è destinatario di chat	User (0, N) Chat(1, 1)	-
Mittente	User è mittente di Chat	User (0, N) Chat(1, 1)	-
Possiede	Utente possiede Sconto	Utente (0, N) Sconto (1, N)	-

3. Vincoli di Integrità:

1. Il contenuto dell'identificativo di User: (Rider, Utente, Ristornate), deve essere rispettivamente: un codice numerico, un'e-mail, e un nome di formato stringa a seconda del tipo rispettivo;
2. Il rider che consegna l'ordine deve essere quello libero con somma minima delle distanze ristornate utente;
3. Il rider che consegna non deve essere un rider in bici o in monopattino per somma distanze ristornate utente maggiore a 10km;
4. La posizione del rider deve essere aggiornata periodicamente;
5. Il rider in monopattino non deve consegnare oltre la sua autonomia;(somma minima delle distanze ristorante utente non superiore a KM rimanenti del rider);
6. Uno fra mittente o destinatario dell'entità chat deve avere obbligatoriamente il tipo user uguale a "utente";
7. Il valore del borsellino deve essere aggiornato dopo ogni ordinazione;
8. L'ordine, per essere inserito, deve avere un costo (ricavato dall'operazione su quantità dettagli ordine e prezzo del piatto e costo spedizione) inferiore o uguale al borsellino dell'utente che lo effettua;
9. L'attributo Top Partner (corrispondente alla data di entrata nella categoria), deve essere modificato solo se il ristornate ha questo attributo uguale a NULL e una percentuale di ordini annullati non superiore al 1.5% degli ordini passati, di ordini con reclamo, non superiore all'2.5% degli ordini passati, inoltre deve avere un numero di ordini passati superiore a 20; (va messo a NULL se non rispetta una o tutte le precedenti condizioni)
10. La categoria del ristorante deve corrispondere alle categorie dei piatti che ha inserito;
11. Lo sconto deve essere rimosso quando viene utilizzato e decurtare dal costo dell'ordine (ricavato dall'operazione su quantità dettagli ordine e prezzo del piatto) la percentuale contenuta in valore;
12. Lo Stato dell'ordine, deve passare a "consegnato" solo quando venga consegnato, ovvero Il timestamp di consegna sia diversa da NULL.
13. Tutte le associazioni specificato tra ordine e piatto ordine corrispondenti a un ordine specifico, devono essere composte da piatti di un unico ristorante;
14. Lo User che scrive la valutazione deve avere l'identificativo uguale a quello di riferimento ordine;
15. Le valutazioni devono valutare esclusivamente user con identificativi ricavabili dall'ordine;
16. La valutazione deve essere scritta da user con tipo utente e valutare user con tipo diverso da utente;
17. Quando il ristornate annulla un ordine deve essere incrementato di uno il suo attributo n° ordini annullati e lo stato dell'ordine deve essere "annullato da ristorante";
18. L'ordine può passare a stato "annullato" esclusivamente se l'ordine è in stato di preparazione;
19. La Chat con attributo tipo uguale a "reclamo" deve riferirsi esclusivamente a ordini già consegnati, e può essere inserita esclusivamente fino a 14 giorni dopo la consegna;
20. La Chat con attributo tipo uguale a "info" deve riferirsi esclusivamente a ordini non consegnati e viene rimossa al momento della consegna;
21. Quando viene aperta per la prima volta una chat con tipo "reclamo" si incrementa il n° di ordini reclamo del ristorante da cui è stato effettuato l'ordine; (identificatore ristorante che compare all'interno di piatto);
22. Al primo ordine di un utente, la consegna non viene sommata al costo dell'ordine;
23. Quando viene effettuata la ricerca per filtro devono comparire esclusivamente i ristoranti che hanno piatti in quella rispettiva categoria;

4. Derivazioni:

1. Il voto di Rider viene calcolato facendo la media dell'attributo voto, nelle occorrenze di valutazione che valutano uno user con identificativo uguale al suo;
2. Il voto di Ristornate viene calcolato facendo la media dell'attributo voto, nelle occorrenze di valutazione che valutano uno user con identificativo uguale al suo.

*Per N° Ord. Reclamo e per N° Ord. Ann. Da Ristorante è presente una regola aziendale specifica.
(Regola 21 e Regola 17)*

Progettazione Logica:

1. Tavola dei Volumi:

Concetto	Tipo	Volume	Giustificazione
Utente	E	10.000	Stima ipotizzata per citta come Torino sui dati di Glovo.it
Ristorante	E	200	Stima ipotizzata per citta come Torino sui dati di Glovo.it
Rider	E	100	Stima ipotizzata per carico di 10 ordini al gg per rider
Rider Monopattino	E	20	Ipotizzato 1/5 dei rider in totali
User	E	10.300	Somma di Utenti, Ristoranti , Rider
Ordine	E	360.000	Totale ordini ipotizzando 3 ordini al mese per utente, 1000 del girono corrente
Ordine passato	E	359.000	Ordini passati, memorizzati per anno
Chat	E	40.000	(C. info * 1gg + C. Reclamo * 14gg)* 10 mess. (chat info 50%, 25% chat rec. Su tot ordini)
Valutazione	E	120.000	Stima ipotizzata per 1/3 degli ordini consegnati recensiti, tra Rider e Ristorante
Metodo di Pagamento	E	15.000	½ metodi di pagamento a utente
Categoria	E	50	Ipotizziamo molte categoria per facilitare ricerca
Sconto	E	20.000	Stima ipotizzata per due sconti ad utente
Piatto	E	2.500	12/13 piatti per ristornate
Utilizza	R	15.500	Sicuramente utilizzato da un utente, ma può essere utilizzato da n utenti
Effettua	R	360.000	Gli ordini effettuati in un anno
Scrive	R	120.000	Valutazioni scritte in un anno
Cucina	R	2.500	Piatti cucinati da ristorante
Valuta	R	120.000	Ricavato dalla sua entità
Consegna	R	360.000	Ricavato dalla sua entità
Specificato	R	1.000.000	Media di 2/3 piatti per ordine

Categorizza	R	9.000	3/4 categorie per piatto
Riferimento	R	120.000	Due valutazioni possono riferirsi allo stesso ordine, ma ogni valutazione ha minimo un ordine di rif.
Riferimento Chat	R	40.000	Una chat deve far riferimento a uno e un solo ordine, ma un ordine può riferirsi a più chat
Destinatario	R	40.000	Identifica esternamente chat quindi devono essercene il num. di chat
Mittente	R	40.000	Identifica esternamente chat quindi devono essercene il num. di chat
Possiede	R	25.000	Lo sconto deve essere posseduto almeno da un utente, ma può essere di più utenti

2. Operazioni più significative:

Individuazione delle operazioni più significative sulla base della regola 80-20;

Operazione	Tipo	Frequenza
1. Inserimento di un nuovo Utente	i	50 / gg
2. Recupero informazioni Ristoranti	i	3.000 /gg
3. Recupero informazioni Ristorante e Piatti	i	10.000 / gg
4. Inserimento di un nuovo ordine	i	1.000 / gg
5. Assegnamento del rider	b	1.000 / gg
6. Agg. Top Partner per ogni Rist. (<i>per analisi ridondanze</i>)	b	1 / settimana

- 3000 /gg: sono 1.000 utenti *3 volte al gg
- 10.000 /gg: sono 1000 *3 volte al gg *3 ristoranti

3. Analisi delle Ridondanze:

Ridondanze in Attributi:

1. N° Ordini Reclamati;
2. Voto Ristorante;
3. Voto Rider;
4. N° ordini Annullati da Rist.

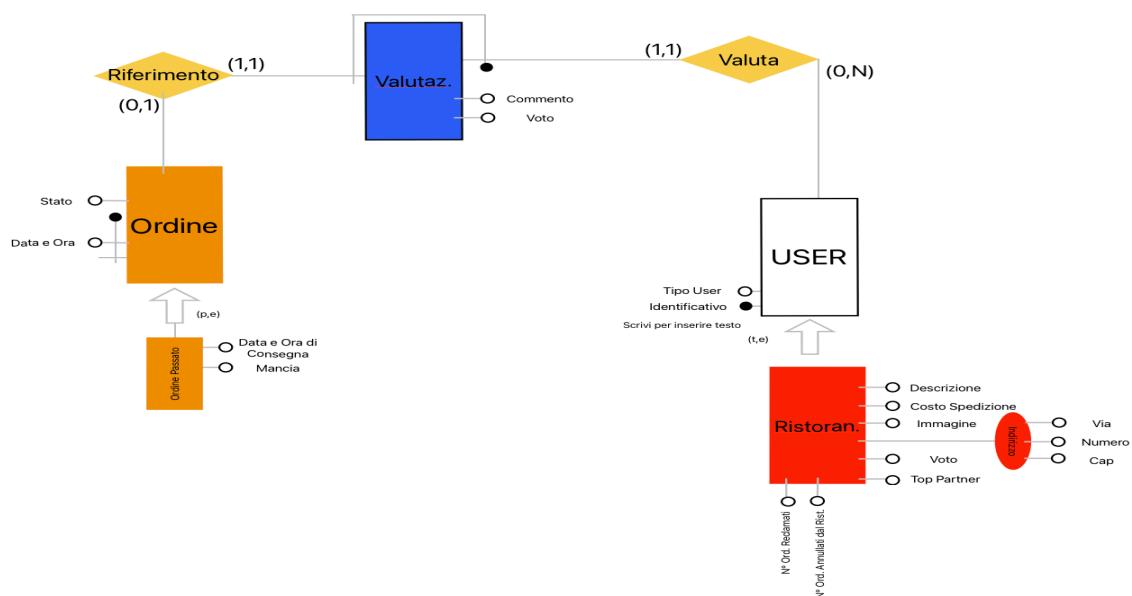
1. Analisi dettagliata: Voto Ristorante:

Principali operazioni coinvolte:

1. Recupero informazioni Ristoranti;
(Voto, Nome, Immagine, Descrizione, N° valutazioni)
2. Aggiornamento di Top Partner;
(Voto, N° ordini annullati, N° reclamo, N° ordini tot del rist.)

Recupero Informazioni Ristoranti:

-Schema di navigazione:



Dettaglio da schema E-R non ristrutturato per il recupero del voto;

-Tavola degli accessi con ridondanza: Recupero Informazioni Ristoranti:

Totale/operazione: $1 + 500 = 501$ accessi;

Concetto	Costrutto	Accessi	Tipo
Ristorante	E	1	L
Valuta	R	500	L

Totale/giorno: 300.600.000; $((200 \text{ rist} * 501 \text{ accessi}) * 3000/\text{gg})$;
-200 ristoranti, -500 valutazioni medie per ristorante in un anno, -3000 ripetizioni gg;

-Tavola degli accessi senza ridondanza: Recupero Informazioni Ristoranti:

Concetto	Costrutto	Accessi	Tipo
Ristorante	E	1	L
Valuta	R	500	L
Valutazione	E	500	L

Totale/operazione: $1+500+500 = 1001$ accessi;

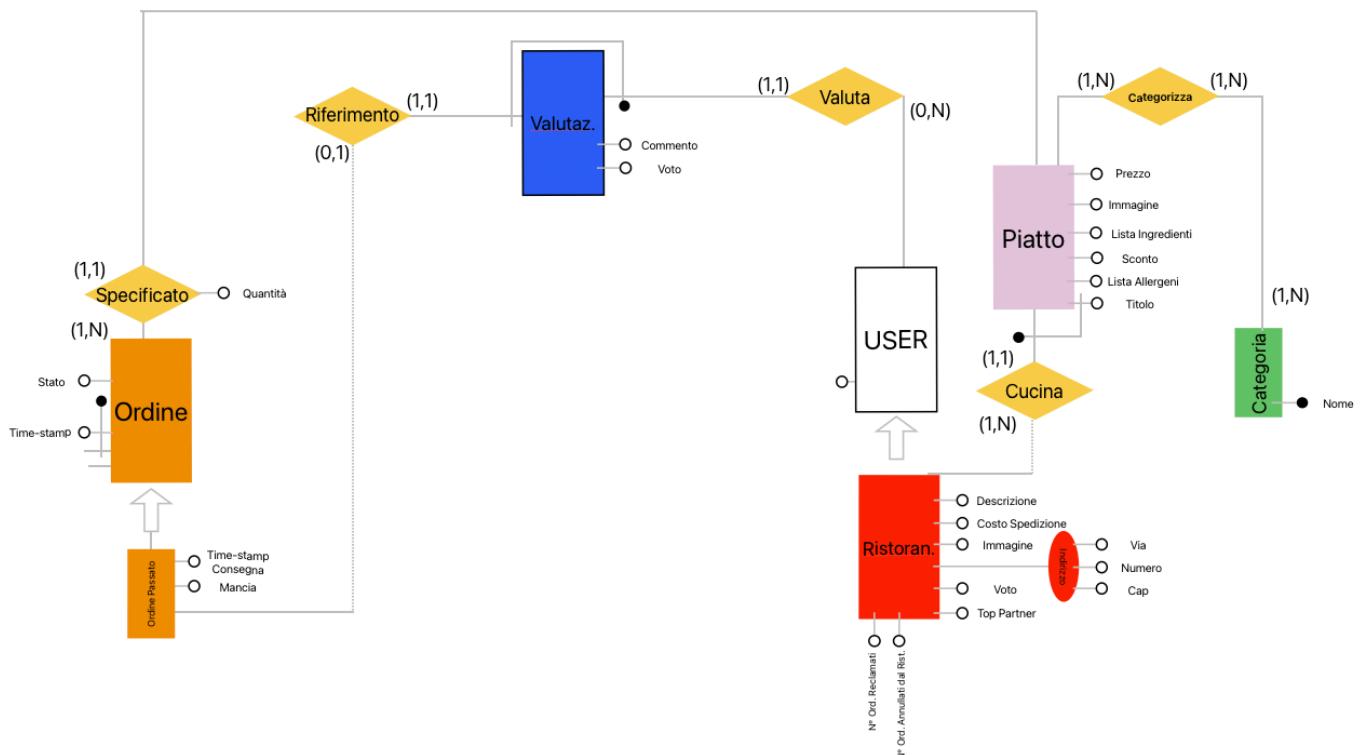
*Totale/giorno: $600.600.000$; $((200 \text{ rist.} * 1001 \text{ accessi}) * 3000/\text{gg})$;*

-200 ristoranti, -500 valutazioni medie per ristorante in un anno, -3000 ripetizioni gg;

NOTA: La ridondanza evita un numero di accessi elevatissimo, anche nell'operazione di aggiornamento Top Partner ci evita moltissimi accessi inutili;

Aggiornamento del Top Partner:

-Schema di navigazione: Aggiornamento del Top partner:



Dettaglio da schema E-R non ristrutturato per il recupero del voto e numero ordini del ristorante;

-Tavola degli accessi con ridondanza: Calcolo del Top Partner:

Si effettua il calcolo di Top Partner, siccome l'interesse dell'esame è valutare la convenienza di Voto Ristorante, per il calcolo si utilizza la ridondanza N° ordini Reclamati e N° ordini annullati da Ristorante per semplicità di rappresentazione.

Concetto	Costrutto	Accessi	Tipo
Ristorante	E	1	L
Cucina	R	12	L
Piatto	E	12	L
Specificato	R	1825	L

Dobbiamo arrivare a specificato per contare il numero di ordini cucinati dal ristorante per verificare che sia superiore a venti

Totale/operazione: $1+12+12+1825=1850$ accessi per 1 rist. / una settimana;

*Totale/settimana: 370.000 accessi; (1850 accessi * 200rist);*

*-1 ristorante, -12 piatti cucinati, -12 piatti,
-1825 n. ordini medio del rist. / anno;*

-Tavola degli accessi senza ridondanza: Calcolo del Top Partner:

Concetto	Costrutto	Accessi	Tipo
Ristorante	E	1	L
Cucina	R	12	L
Piatto	E	12	L
Specificato	E	1825 (ordini singoli)	L
Valuta	R	500	L
Valutazione	E	500	L

Totale/operazione: $1+12+12+1825+500+500=2850$ accessi per 1 rist. / una sett.;

*Totale/settimana: 570.000 accessi; (2850 accessi * 200rist);*

*-1 ristorante, -12 piatti cucinati, -12 piatti,
-1825 n. ordini medio del rist. / anno;*

-Tavola dei costi in termini di spazio: Recupero Informazioni Ristoranti:

Si ipotizza 4 byte, (Real) siccome il valore di voto Ristorante è numerico da 1 a 5, ma deve considerare il numero dopo la virgola per la media. Moltiplicandolo per le 200 occorrenze di ristorante, si ottengono 800 Byte < 1kByte, quindi uno spazio trascurabile.

Ridondanza	Senza Ridondanza
301.000.000 accessi	601.200.000 accessi
<1kByte aggiuntivo	0 kByte aggiuntivo

-Come numero di accessi si è calcolata la somma fra il numero di accessi delle due operazioni principali che coinvolgono la ridondanza Voto Ristorante.

Conclusione: E' conveniente mantenere la ridondanza perché ci evitiamo la metà degli accessi;

4. Eliminazione delle Generalizzazioni:

1. Generalizzazione Rider -> Rider Monopattino; (Parziale, Esclusiva)
2. Generalizzazione User -> Rider; (Totale, Esclusiva)
-> Ristorante;
-> Utente;
3. Generalizzazione Ordini -> Ordini Passati; (Parziale, Esclusiva)

Soluzioni ipotizzate per l'eliminazione:

-Accorpamento di Rider Monopattino nel genitore Rider:

Spreco per valori nulli ma riduzione degli accessi durante le query;

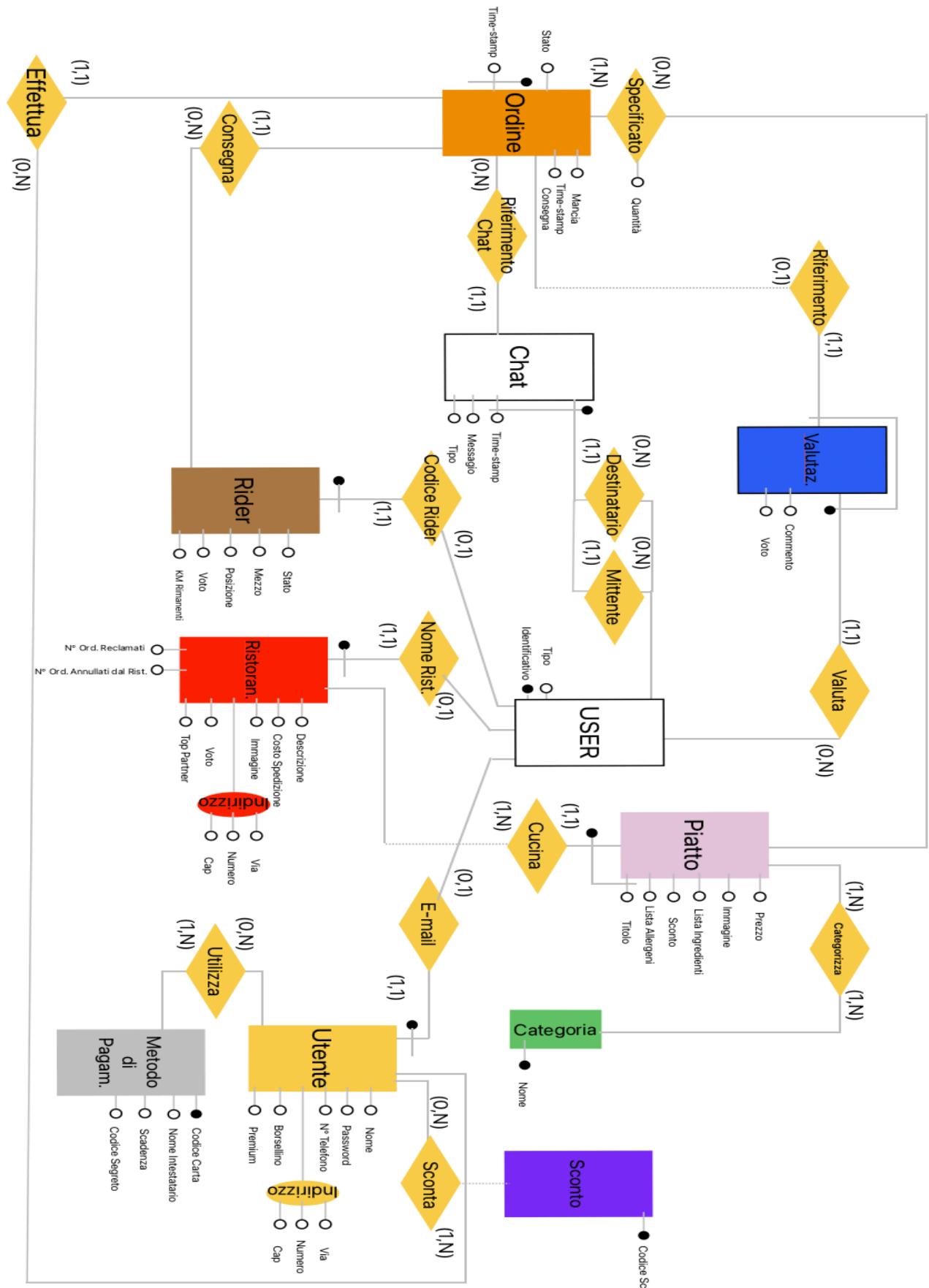
-Sostituzione della generalizzazione con delle relazioni:

Applicabile poiché tutte le entità figlie presentano accessi distinti; e perché la generalizzazione è totale;

-Accorpamento di Ordine Passato nel genitore Ordine:

Spreco per valori nulli ma riduzione degli accessi durante le query;

5. Schema E-R Ristrutturato:



6. Commento:

1. Commento schema ER Ristrutturato:

E' stata eseguita la ristrutturazione dello schema ER non ristrutturato, seguendo le slide proposte a lezione, quindi eliminando le generalizzazioni seguendo il tipo di correzione previsto per ogni cardinalità.

2. Accorpamento di Entità o Relazioni:

Durante la ristrutturazione, è stata individuata una entità superflua: Dettaglio Ordine. La quale si associava a piatto attraverso una relazione 1 a 1, quindi inutile, ed è stata rimossa già dallo schema ER non ristrutturato e dai requisiti rivisti, accorpandola a "specificato", associazione di ordine con piatto.

3. Scelta degli identificatori principali:

Nella ristrutturazione sono stati mantenuti gli identificatori richiesti, trovati nella progettazione concettuale dello schema non ristrutturato, senza introdurne nuovi;

7. Regole Aziendali: Vincoli di Integrità:

1. Il contenuto dell'identificativo di User: (Rider, Utente, Ristornate), deve essere rispettivamente: un codice numerico, un'e-mail, e un nome di formato stringa a seconda del tipo rispettivo;
2. Il rider che consegna l'ordine deve essere quello libero con somma minima delle distanze ristorante utente;
3. Il rider che consegna non deve essere un rider in bici o in monopattino per somma distanze ristorante utente maggiore a 10km;
4. La posizione del rider deve essere aggiornata periodicamente;
5. Il rider in monopattino non deve consegnare oltre la sua autonomia; (somma minima delle distanze ristorante utente non superiore a KM rimanenti del rider);
6. Uno fra mittente o destinatario dell'entità chat deve avere obbligatoriamente il tipo di user uguale a "utente";
7. Il valore del borsellino deve essere aggiornato dopo ogni ordinazione;
8. L'ordine, per essere inserito, deve avere un costo (ricavato dall'operazione su quantità dettagli ordine e prezzo del piatto e costo spedizione) inferiore o uguale al borsellino dell'utente che lo effettua;
9. L'attributo Top Partner (corrispondente alla data di entrata nella categoria), deve essere modificato solo se il ristornate ha questo attributo uguale a NULL e una percentuale di ordini annullati non superiore al 1.5% degli ordini passati, di ordini con reclamo, non superiore all'2.5% degli ordini passati, inoltre deve avere un numero di ordini passati superiore a 20; (va messo a NULL se non rispetta una o tutte le precedenti condizioni)
10. La categoria del ristorante deve corrispondere alle categorie dei piatti che ha inserito;
11. Lo sconto deve essere rimosso quando viene utilizzato e decurtare dal costo dell'ordine (ricavato dall'operazione su quantità dettagli ordine e prezzo del piatto) la percentuale contenuta in valore;
12. Lo Stato dell'ordine, deve passare a "consegnato" solo quando venga consegnato, ovvero Il time-stamp di consegna sia diversa da NULL.
13. Tutte le associazioni specificate tra ordine e piatto ordine corrispondenti a un ordine specifico, devono essere composte da piatti di un unico ristorante;
14. Lo User che scrive la valutazione deve avere l'identificativo uguale a quello di riferimento ordine;
15. Le valutazioni devono valutare esclusivamente user con identificativi ricavabili dall'ordine;

16. La valutazione deve essere scritta da user con tipo “utente” e valutare user con tipo diverso da “utente”;
17. Quando il ristorante annulla un ordine deve essere incrementato di uno il suo attributo n° ordini annullati, e nello stato dell’ordine va inserito “annullato da ristorante”;
18. L’ordine può passare a stato “annullato” esclusivamente se l’ordine è in stato di preparazione;
19. La Chat con attributo tipo uguale a “reclamo” deve riferirsi esclusivamente a ordini già consegnati, e può essere inserita esclusivamente fino a 14 giorni dopo la consegna;
20. La Chat con attributo tipo uguale a “info” deve riferirsi esclusivamente a ordini non consegnati e viene rimossa al momento della consegna;
21. Quando viene aperta per la prima volta una chat con tipo “reclamo” si incrementa il n° di ordini reclamo del ristorante da cui è stato effettuato l’ordine; (identificatore ristorante che compare all’interno di piatto);
22. Al primo ordine di un utente, la consegna non viene sommata al costo dell’ordine;
23. Quando viene effettuata la ricerca per filtro devono comparire esclusivamente i ristoranti che hanno piatti in quella rispettiva categoria;

Traduzione Concettuale - Logica:

1. Schema Logico:

User (Identificativo, Tipo);

Utente (E-mail, Nome, Password, N° Telefono, Indirizzo, Borsellino, Premium);

Ristorante (Nome, Descrizione, Costo spedizione, Immagine, Voto, Indirizzo, Top-Partner, n° ordini annullati, n° ordini reclamo);

Rider (Codice Rider, Stato, Mezzo, Posizione, Voto, (KM Rimanenti se in monopattino));

Metodo di Pagamento (Codice carta, Nome intestatario, Scadenza, Codice Segreto);

Utilizza (E-mail, Codice carta);

Sconto (Codice sconto, valore);

Sconta (E-mail, Codice sconto);

Categorizza (Titolo Piatto, Nome Ristorante, Nome Categoria);

Categoria (Nome);

Piatto (Titolo, Nome ristorante, Prezzo, Immagine, Lista allergeni, Lista ingredienti, Sconto);

Chat (Destinatario, Mittente, Data e ora, Email_Utente, Time-Stamp_ordine, Messaggio);

Ordine (E-mail Utente, Time-stamp, Codice Rider, Stato, Mancia*, Time-stamp Consegnna*);

Specificato (Titolo Piatto, Nome Ristorante, E-mail utente, Time-stamp, Quantità);

Valutazione (E-mail Utente, Time-stamp Ordine, Identificativo valutato, Commento, Voto);

2. Vincoli di integrità referenziale:

3. E-mail in utente e (la chiave di) User;
4. Codice rider in Rider e (la chiave di) User;
5. Nome ristorante in Ristorante e (la chiave di) User;
6. E-mail in Utilizza e (la chiave di) Utente;
7. Codice carta in Utilizza e (la chiave di) Metodo di Pagamento;
8. E-mail in Sconta e (la chiave di) Utente;
9. Codice sconto in Sconta e (la chiave di) Sconto;
10. Titolo piatto, nome ristorante in Categorizza e (la chiave di) Piatto;
11. Nome categoria in Categorizza (la chiave di) Categoria;
12. Nome ristorante in Piatto e (la chiave di) Ristorante;
13. Destinatario, Mittente in chat e (la chiave di) User;
14. E-mail utente in Ordine e (la chiave di) Utente;
15. Titolo piatto in Specificato e (la chiave di) Piatto;
16. Nome ristorante in Specificato e (la chiave di) Ristorante;
17. E-mail utente, time-stamp in Specificato e (la chiave di) Ordine;
18. E-mail utente, time-stamp ordine in Valutazione e (la chiave di) Ordine;
19. Identificativo valutato in Valutazione e (la chiave di) User;

DDL:

```
CREATE TABLE UserBase (
    Identificativo VARCHAR(255) PRIMARY KEY,
    Tipo VARCHAR(50) NOT NULL
);

CREATE TABLE Utente (
    Email VARCHAR(255) PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    N_Telefono VARCHAR(15) NOT NULL,
    Indirizzo VARCHAR(255) NOT NULL,
    Borsellino DECIMAL(10, 2) DEFAULT 0,
    Premium BOOLEAN DEFAULT false,
    CONSTRAINT fk_utente_user FOREIGN KEY (Email) REFERENCES UserBase(Identificativo) ON
    UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Ristorante (
    Nome_Ristorante VARCHAR(255) PRIMARY KEY,
    Descrizione TEXT,
    Costo_spedizione DECIMAL(5, 2) DEFAULT 0,
    Immagine TEXT,
    Voto REAL CHECK (Voto >= 0 AND Voto <= 5),
    Indirizzo VARCHAR(255) NOT NULL,
    Top_Partner DATE DEFAULT NULL,
    n_ordini_annullati INT DEFAULT 0,
    n_ordini_reclamo INT DEFAULT 0,
    CONSTRAINT fk_ristorante_user FOREIGN KEY (Nome_Ristorante) REFERENCES
    UserBase(Identificativo) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Rider (
    Codice_rider VARCHAR(255) PRIMARY KEY,
    Stato VARCHAR(50) NOT NULL,
    Mezzo VARCHAR(50) NOT NULL,
    Posizione VARCHAR(255),
    Voto REAL CHECK (Voto >= 0 AND Voto <= 5),
    KM_Rimanenti DECIMAL(5, 2) DEFAULT NULL,
    CONSTRAINT fk_rider_user FOREIGN KEY (Codice_rider) REFERENCES UserBase(Identificativo)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

```

CREATE TABLE Metodo_di_Pagamento (
    Codice_carta VARCHAR(16) PRIMARY KEY,
    Nome_intestatario VARCHAR(100) NOT NULL,
    Scadenza DATE NOT NULL,
    Codice_Segreto CHAR(3) NOT NULL
);

CREATE TABLE Utilizza (
    Email VARCHAR(255),
    Codice_carta VARCHAR(16),
    CONSTRAINT pk_Utilizza PRIMARY KEY (Email, Codice_carta),
    CONSTRAINT fk_utilizza_Email FOREIGN KEY (Email) REFERENCES Utente(Email) ON UPDATE
    CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_utilizza_carta FOREIGN KEY (Codice_carta) REFERENCES
    Metodo_di_Pagamento(Codice_carta) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Sconto (
    Codice_sconto VARCHAR(50) PRIMARY KEY,
    Valore_percentuale SMALLINT NOT NULL CHECK (Valore_percentuale > 0 AND
    Valore_percentuale <= 100)
);

CREATE TABLE Sconta (
    Email VARCHAR(255),
    Codice_sconto VARCHAR(50),
    CONSTRAINT pk_Sconta PRIMARY KEY (Email, Codice_sconto),
    CONSTRAINT fk_Sconta_Email FOREIGN KEY (Email) REFERENCES Utente(Email) ON UPDATE
    CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_Sconta_Sconto FOREIGN KEY (Codice_sconto) REFERENCES
    Sconto(Codice_sconto) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Piatto (
    Titolo VARCHAR(255),
    Nome_Ristorante VARCHAR(255),
    Prezzo DECIMAL(5, 2) NOT NULL,
    Immagine TEXT,
    Lista_allergeni TEXT NOT NULL,
    Lista_ingredienti TEXT NOT NULL,
    Sconto SMALLINT NOT NULL CHECK (Sconto >= 0 AND Sconto <= 100) DEFAULT 0,
    CONSTRAINT pk_Pitto PRIMARY KEY (Titolo, Nome_Ristorante),
    CONSTRAINT fk_Ristorante_Cucina FOREIGN KEY (Nome_Ristorante) REFERENCES
    Ristorante(Nome_Ristorante) ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE Categoria (
    Nome VARCHAR(100) PRIMARY KEY
);

CREATE TABLE Categorizza (
    Titolo_Piatto VARCHAR(255),
    Nome_Ristorante VARCHAR(255),
    Nome_Categoria VARCHAR(100),
    CONSTRAINT pk_Categorizza PRIMARY KEY (Titolo_Piatto, Nome_Ristorante, Nome_Categoria),
    CONSTRAINT fk_Categorizza_Piatto FOREIGN KEY (Titolo_Piatto, Nome_Ristorante) REFERENCES
Piatto(Titolo, Nome_Ristorante) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_Categorizza_Categoria FOREIGN KEY (Nome_Categoria) REFERENCES
Categoria(Nome) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Ordine (
    Email_Utente VARCHAR(255),
    Time_stamp TIMESTAMP,
    Codice_Rider VARCHAR(255),
    Stato VARCHAR(50) DEFAULT 'In preparazione',
    Mancia DECIMAL(3, 1) DEFAULT 0,
    Time_stamp_Consegna TIMESTAMP,
    CONSTRAINT pk_Ordine PRIMARY KEY (Email_Utente, Time_stamp),
    CONSTRAINT fk_ordine_utente FOREIGN KEY (Email_Utente) REFERENCES Utente>Email
ON
UPDATE CASCADE,
    CONSTRAINT fk_ordine_rider FOREIGN KEY (Codice_Rider) REFERENCES Rider(Codice_rider) ON
UPDATE CASCADE ON DELETE SET NULL
);

CREATE TABLE Specificato (
    Titolo_Piatto VARCHAR(255),
    Nome_Ristorante VARCHAR(255),
    Email_Utente VARCHAR(255),
    Time_stamp TIMESTAMP,
    Quantità SMALLINT CHECK (Quantità > 0) DEFAULT 1,
    CONSTRAINT pk_Specificato PRIMARY KEY (Titolo_Piatto, Nome_Ristorante, Email_Utente,
Time_stamp),
    CONSTRAINT fk_Specificato_Piatto FOREIGN KEY (Titolo_Piatto, Nome_Ristorante) REFERENCES
Piatto(Titolo, Nome_Ristorante) ON UPDATE CASCADE,
    CONSTRAINT fk_Specificato_Ordine FOREIGN KEY (Email_Utente, Time_stamp) REFERENCES
Ordine>Email_Utente, Time_stamp
ON UPDATE CASCADE ON DELETE CASCADE
);

```

```

CREATE TABLE Chat (
    Destinatario VARCHAR(255),
    Mittente VARCHAR(255),
    Data_e_ora TIMESTAMP,
    Messaggio TEXT NOT NULL,
    Timestamp_ordine TIMESTAMP,
    Email_Utente varchar(255),
    Tipo varchar(50),
    CONSTRAINT pk_Chat PRIMARY KEY (Destinatario, Mittente, Data_e_ora),
    CONSTRAINT fk_Chat_Destinatario FOREIGN KEY (Destinatario) REFERENCES
UserBase(Identificativo) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_Chat_Mittente FOREIGN KEY (Mittente) REFERENCES UserBase(Identificativo)
ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_Chat_Riferimento FOREIGN KEY (Timestamp_ordine,Email_Utente)
REFERENCES Ordine(Time_stamp, Email_Utente) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Valutazione (
    Email_Utente VARCHAR(255),
    Time_stamp_Ordine TIMESTAMP,
    Identificativo_valutato VARCHAR(255),
    Commento TEXT,
    Voto SMALLINT CHECK (Voto >= 0 AND Voto <= 5),
    CONSTRAINT pk_Valutazione PRIMARY KEY (Email_Utente, Time_stamp_Ordine,
Identificativo_valutato),
    CONSTRAINT fk_valutazione_ordine FOREIGN KEY (Email_Utente, Time_stamp_Ordine)
REFERENCES Ordine(Email_Utente, Time_stamp) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_valutazione_user FOREIGN KEY (Identificativo_valutato) REFERENCES
UserBase(Identificativo) ON UPDATE CASCADE ON DELETE CASCADE
);

```

1.Commento DDL:

Al fine del funzionamento su PgAdmin4 la relazione dello schema logico di nome User è stata modificata con il nome UserBase, poiché User è una keyword non utilizzabile in Postgress.

DML:

-- Inserimento di un nuovo utente

```
INSERT INTO UserBase (Identificativo, Tipo) VALUES ('utente1@example.com', 'Utente');
INSERT INTO Utente (Email, Nome, Password, N_Teléfono, Indirizzo, Borsellino, Premium)
VALUES ('utente1@example.com', 'Mario Rossi', 'password123', '1234567890', 'Via Roma 1,
Milano', 0.0, false);
```

-- Inserimento di un nuovo ristorante

```
INSERT INTO UserBase (Identificativo, Tipo) VALUES ('ristorante1', 'Ristorante');
INSERT INTO Ristorante (Nome_Ristorante, Descrizione, Costo_spedizione, Immagine, Indirizzo)
VALUES ('Ristorante da Giovanni', 'Cucina italiana tradizionale', 2.0, 'path_immagine.jpg','Via
Garibaldi 2, Roma');
```

-- Inserimento di un nuovo rider

```
INSERT INTO UserBase (Identificativo, Tipo) VALUES ('rider1', 'Rider');
INSERT INTO Rider (Codice_rider, Stato, Mezzo, Posizione)
VALUES ('rider1', 'Disponibile', 'Bicicletta', 'Via Venezia 3, Milano');
```

-- Inserimento di un nuovo metodo di pagamento e associazione ad un utente

```
INSERT INTO Metodo_di_Pagamento (Codice_carta, Nome_intestatario, Scadenza,
Codice_Segreto)
VALUES ('1234567890123456', 'Mario Rossi', '2026-12-31', '123');
INSERT INTO Utilizza (Email, Codice_carta) VALUES ('utente1@example.com',
'1234567890123456');
```

-- Inserimento di un nuovo sconto e associazione ad un utente

```
INSERT INTO Sconto (Codice_sconto, Valore_percentuale) VALUES ('SUMMER2024', 20);
INSERT INTO Sconta (Email, Codice_sconto) VALUES ('utente1@example.com', 'SUMMER20');
```

-- Inserimento di un nuovo piatto e inserimento categoria

```
INSERT INTO Piatto (Titolo, Nome_Ristorante, Prezzo, Immagine, Lista_allergeni, Lista_ingredienti,
Sconto)
VALUES ('Pizza Margherita', 'Ristorante da Giovanni', 7.50, 'path_immagine.jpg', 'Glutine',
'Pomodoro, mozzarella, basilico', 0);
```

-- Inserimento di una nuova categoria

```
INSERT INTO Categoria (Nome) VALUES ('Pizza');
```

-- Associazione tra un piatto inserito ed una categoria

```
INSERT INTO Categorizza (Titolo_Piatto, Nome_Ristorante, Nome_Categoria)
VALUES ('Pizza Margherita', 'Ristorante da Giovanni', 'Pizza');
```

```
-- Creazione di un nuovo ordine
INSERT INTO Ordine (Email_Utente, Time_stamp, Codice_Rider, Stato, Mancia)
VALUES ('utente1@example.com', '2024-06-06 12:00:00', 'rider1', 'In preparazione', 0.0);
-- Inserimento di specifiche degli articoli nell'ordine
INSERT INTO Specificato (Titolo_Piatto, Nome_Ristorante, Email_Utente, Time_stamp, Quantità)
VALUES ('Pizza Margherita', 'Ristorante da Giovanni', 'utente1@example.com', '2024-06-06
12:00:00', 2);

-- Inserimento di una chat per informazioni sull'ordine non ancora consegnato
INSERT INTO Chat (Destinatario, Mittente, Data_e_ora, Messaggio, Timestamp_ordine,
Email_Utente, Tipo)
VALUES ('Ristorante da Giovanni', 'utente1@example.com', CURRENT_TIMESTAMP, 'Vorrei sapere
lo stato del mio ordine.', '2024-06-06 12:00:00', 'utente1@example.com', 'Informazioni');
```

Modifiche e Cancellazioni per la verifica dei vincoli:

```
-- Modifica dello stato di un ordine
UPDATE Ordine
SET Stato = 'Consegnato', Time_stamp_Consegna = CURRENT_TIMESTAMP
WHERE Email_Utente = 'utente1@example.com' AND Stato = 'In preparazione' AND time_stamp =
'2024-06-06 12:00:00';

-- Inserimento di una valutazione dell'ordine
INSERT INTO Valutazione (Email_Utente, Time_stamp_Ordine, Identificativo_valutato, Commento,
Voto)
VALUES ('utente1@example.com', '2024-06-06 12:00:00', 'Ristorante da Giovanni', 'Servizio
eccellente!', 5);

--Cancellazione di un rider
DELETE FROM userBase WHERE codice_rider = 'rider1';

--Cancellazione di un ordine
DELETE FROM ordine WHERE email_utente = 'utente1@example.com' AND time_stamp = '2024-
06-06 12:00:00';
```