

Progettazione e sviluppo di una Base di dati relazionale per la gestione di una catena di ristorazione

Fabio Cinicolo
N86003340

Umberto Elias De Angelis
N86003190

Mirco Napolitano
N86003332

Indice

1. Descrizione
2. Class diagram
3. Class diagram ristrutturato
4. Dizionario delle classi
5. Dizionario delle associazioni
6. Dizionario dei vincoli
- 7.

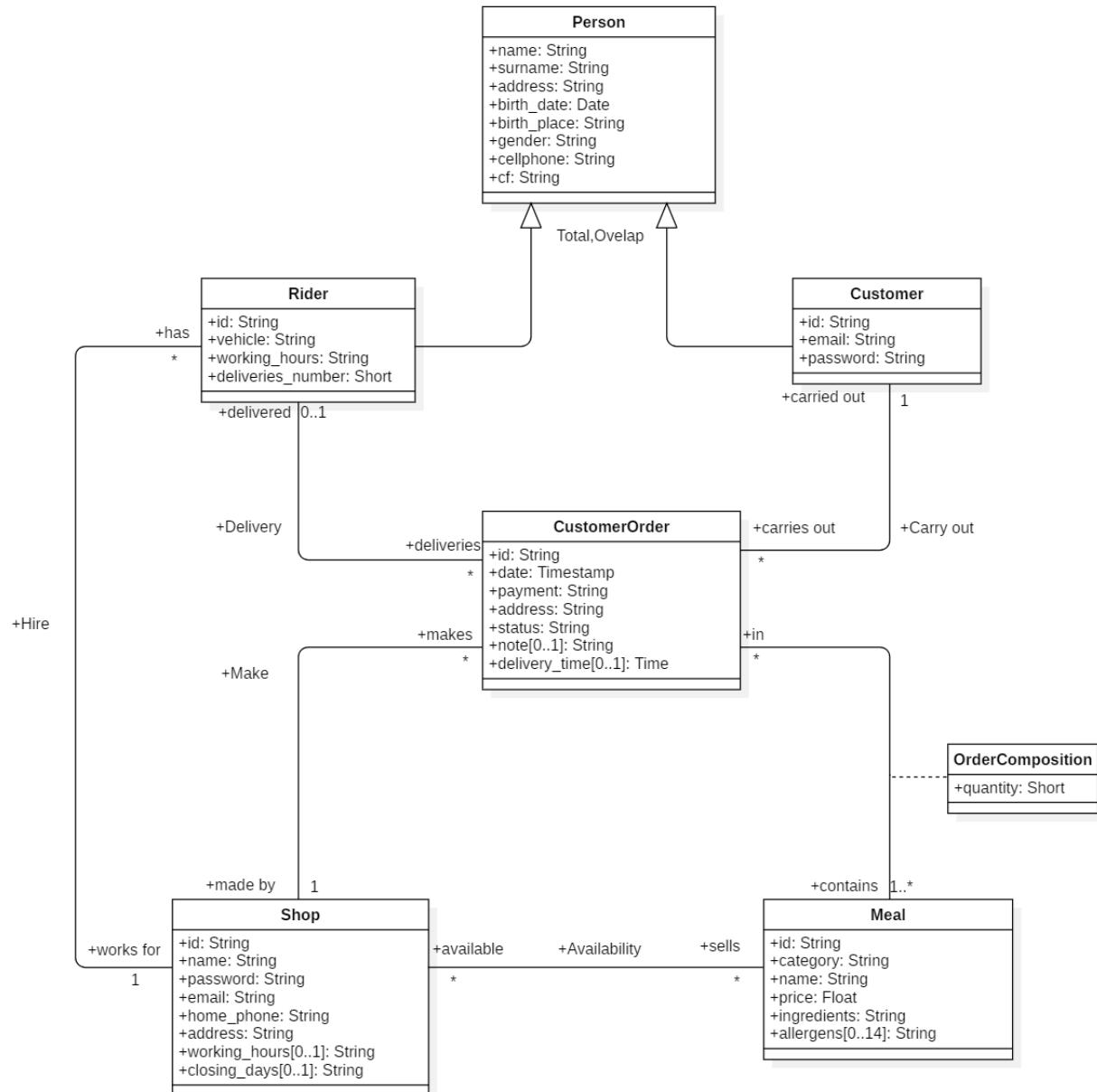
Descrizione

La base di dati è stata pensata e realizzata per soddisfare le esigenze di diverse tipologie di utenti della catena di ristorazione, si è quindi deciso di categorizzarli in:

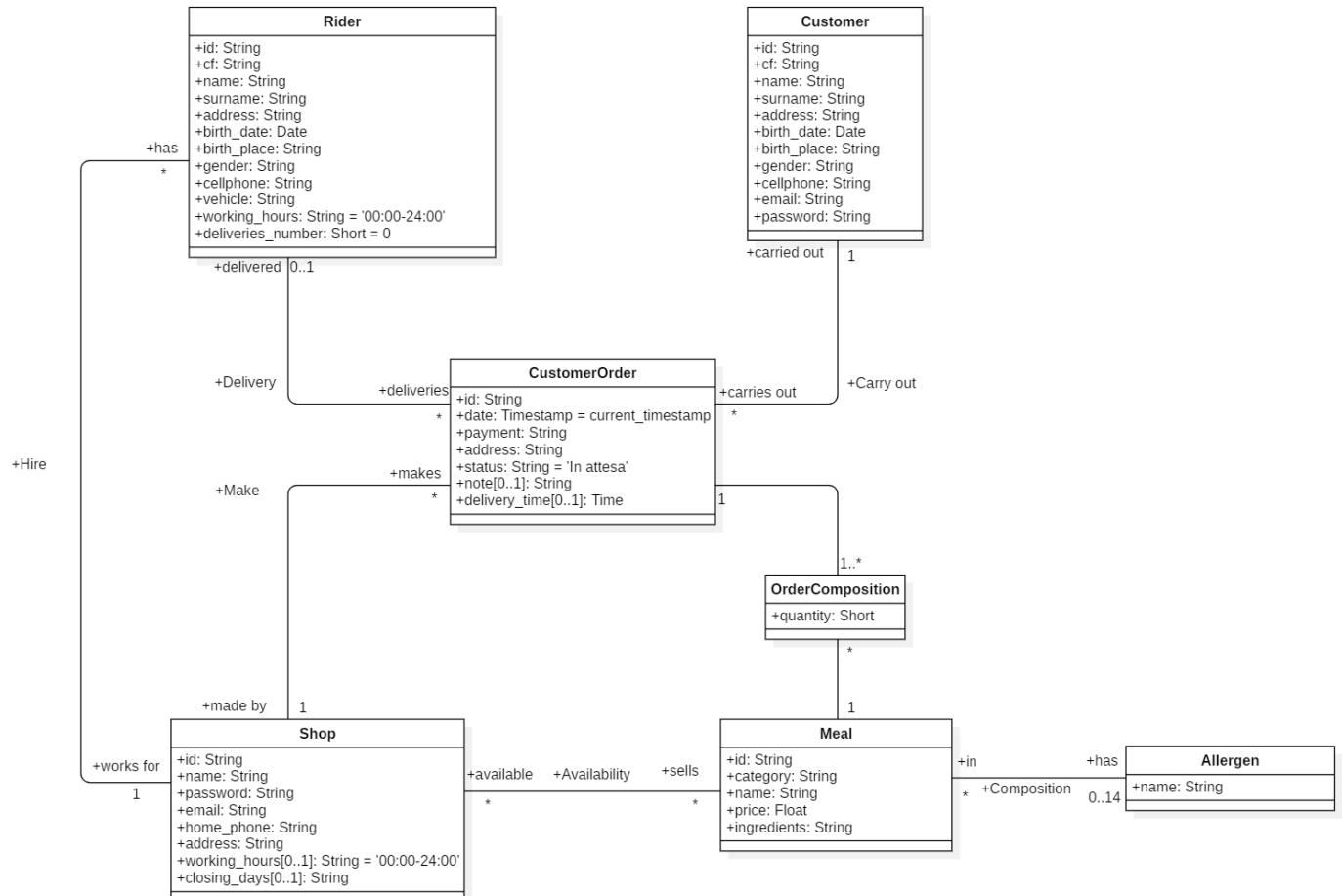
- Clienti : potranno registrarsi ed effettuare l'accesso all'applicativo. Potranno effettuare ordini verso la catena di ristorazione. Inoltre, sarà possibile effettuare una ricerca di alimenti combinando categoria, fascia di prezzo e allergie alimentari.
- Ristoranti : ristoratori che accederanno all'applicativo di gestione del proprio ristorante. Potranno gestire i rider (in particolare assumerne o licenziarne), inserire o rimuovere alimenti della catena di ristorazione nel menù e visualizzarli. Potranno, inoltre, assegnare una consegna ad un rider a patto che non abbia più di tre consegne in corso. Un ordine verrà considerato completato quando questo verrà consegnato al cliente.
- Amministratore : effettuerà l'accesso al pannello di amministrazione della catena di ristorazione, potrà poi visualizzare e gestire i diversi ristoranti, inserire nuovi alimenti, visualizzare la clientela. Potrà inoltre effettuare una ricerca complessa di ordini combinando: veicolo del rider, fascia di prezzo del totale dell'ordine, categoria dei pasti e provincia di consegna. Il software, inoltre, permette al generazione del codice fiscale in fase di registrazione di un nuovo cliente (permettendo la modifica di questo in caso di omocodia dal pannello profilo del cliente) e in fase di assunzione di un rider.

E' possibile visionare un manuale per l'utilizzo del software tramite il seguente link:
[Manuale per l'utilizzo del software Food Overflow](#)

Class Diagram



Class Diagram Ristrutturato



Dizionario delle classi

Classe	Descrizione	Attributi
Customer	Contiene le informazioni personali di un cliente registrato nell'applicativo	<p>id (string): codice identificativo univoco attribuito ad un cliente.</p> <p>cf (string): codice fiscale del cliente.</p> <p>name (string): nome del cliente.</p> <p>surname (string): cognome del cliente.</p> <p>address (string): indirizzo di consegna del cliente.</p> <p>birth_date (date): anno di nascita del cliente.</p> <p>birth_place (string): luogo di nascita del cliente.</p> <p>gender (string): sesso del cliente.</p> <p>cellphone (string): numero di cellulare del cliente.</p> <p>email (string): indirizzo e-mail del cliente.</p> <p>password (string): chiave d'accesso del cliente.</p>
Rider	Contiene le informazioni personali e lavorative di un rider della catena di ristorazione	<p>cf (string): codice fiscale del rider.</p> <p>name (string): nome del rider.</p> <p>surname (string): cognome del rider.</p> <p>address (string): indirizzo di residenza del rider.</p> <p>birth_date (date): anno di nascita del rider.</p> <p>birth_place (string): luogo di nascita del rider.</p> <p>gender (string): sesso del rider.</p> <p>cellphone (string): numero di cellulare del rider.</p> <p>vehicle (string): tipologia di veicolo utilizzato per le consegne.</p> <p>working_hours (string): Fascia oraria lavorativa del rider.</p> <p>deliveries_number (short): numero di consegne in corso (massimo 3).</p>
Shop	Contiene le informazioni di un determinato ristorante della catena di ristorazione	<p>id (string): codice identificativo univoco attribuito ad un ristorante.</p> <p>name (string): nome del ristorante.</p> <p>email (string): indirizzo e-mail del ristorante.</p> <p>home_phone (string): numero di telefono del ristorante.</p> <p>address (string): indirizzo del ristorante.</p> <p>working_hours (string): fascia oraria lavorativa del ristorante.</p> <p>closing_days (string): giorni di chiusura del ristorante.</p>
Meal	Contiene le informazioni di un alimento della catena di ristorazione	<p>id (string): codice identificativo univoco attribuito ad un alimento.</p> <p>category (string): tipologia dell'alimento (carne, pasta, etc.).</p> <p>name (string): nome dell'alimento.</p> <p>price (float): prezzo dell'alimento.</p> <p>ingredients (string): ingredienti di quell'alimento.</p>
Allergen	Contiene i diversi allergeni	name (string): nome di un allergene.
CustomerOrder	Contiene le informazioni di diversi ordini eseguiti da diversi ristoranti della catena di ristorazione	<p>id (string): codice identificativo univoco attribuito ad un ordine.</p> <p>date (timestamp): data di creazione ordine da parte di un cliente.</p> <p>payment (string): tipologia di pagamento selezionato da un cliente.</p> <p>address (string): indirizzo di consegna dell'ordine.</p> <p>status (string): stato dell'ordine (In attesa, In consegna, etc.).</p> <p>note (string): eventuali note lasciate da un cliente in fase di creazione ordine.</p> <p>delivery_time (time): orario in cui lo stato della consegna è diventato "Consegnato" o "Errore".</p>
OrderComposition	Contiene la composizione di alimenti di diversi ordini	quantity (short): quantità dell'alimento dal cliente selezionata in fase di creazione ordine.

Dizionario delle associazioni

Nome	Descrizione	Classi coinvolte
Hire	Esprime il contratto tra un rider ed uno shop.	Rider [*] ruolo (works for): indica il Rider che lavora per lo Shop. Shop [1] ruolo (has): indica lo Shop che assume dei Rider.
Delivery	Esprime l'azione di consegna.	Rider [0..1] ruolo (deliveries): indica il Rider che effettua la consegna. CustomerOrder [*] ruolo (delivered): indica la consegna effettuata dal Rider.
Availability	Esprime la presenza o meno di un Meal nel menù di un negozio.	Shop [*] ruolo (sells): indica lo Shop che dispone degli alimenti. Meal [*] ruolo (available): indica il Meal disponibile per lo Shop.
Composition	Esprime gli allergeni di un alimento.	Meal [*] ruolo (has): indica il Meal che potrà avere allergeni. Allergens [0..14] ruolo (in): indica l'allergene contenuto in un alimento.
Make	Esprime la preparazione degli ordini da parte di un negozio.	CustomerOrder [*] ruolo (made by): indica l'ordine gestito da uno Shop. Shop [1] ruolo (makes): indica lo Shop che prepara l'ordine.
Carry out	Esprime un legame tra l'utente e l'ordine.	Customer [1] ruolo (carried out): indica l'utente che effettua l'ordine. CustomerOrder [*] ruolo (carries out): indica l'ordine effettuato da un utente.

Dizionario dei vincoli

Vincolo	Descrizione
customer_address_check	L'indirizzo del Customer deve avere il seguente formato: “<denominazione urbanistica>, <indirizzo>, <numero civico>, <CAP>, <Comune>, <Provincia>”.
customer_order_address_check	L'indirizzo di consegna deve avere il seguente formato: “<denominazione urbanistica>, <indirizzo>, <numero civico>, <CAP>, <Comune>, <Provincia>”.
rider_address_check	L'indirizzo di residenza del Rider deve avere il seguente formato: “<denominazione urbanistica>, <indirizzo>, <numero civico>, <CAP>, <Comune>, <Provincia>”.
shop_address_check	L'indirizzo dello Shop deve avere il seguente formato: “<denominazione urbanistica>, <indirizzo>, <numero civico>, <CAP>, <Comune>, <Provincia>”.
customer_email_check	L' email del customer deve essere scritta in modo corretto.
shop_email_check	L'email dello Shop deve avere per dominio : “foodoverflow.it”.
customer_cf_check	Il codice fiscale del Customer deve essere scritto in modo legittimo.
rider_cf_check	Il codice fiscale del Rider deve essere scritto in modo legittimo.
shop_closing_days_check	I giorni di chiusura di uno shop devono essere scritti in sequenza, separati tra di loro dai caratteri virgola-spazio.
shop_working_hours_check	L'orario di apertura dello Shop deve essere scritto nel formato specificato (HH:MM-HH:MM).
rider_working_hours_check	L'orario lavorativo del Rider deve essere scritto nel formato specificato (HH:MM-HH:MM).
customer_cellphone_check	Il numero di cellulare del Customer deve essere scritto in modo legittimo, ovvero deve contenere dieci cifre.
rider_cellphone_check	Il numero di cellulare del Rider deve essere scritto in modo legittimo, ovvero deve contenere dieci cifre.
shop_home_phone_check	Il numero di telefono dello Shop deve essere scritto in modo legittimo, ovvero deve contenere un prefisso numerico nazionale.
rider_deliveries_number_check	Il numero di consegne di un Rider deve essere compreso tra zero e tre.
meal_price_check	Il prezzo di un Meal deve essere maggiore di zero.
order_composition_quantity_check	La quantità di un OrderComposition deve essere maggiore di zero
customer_order_status_check	Lo status di un ordine deve essere uno tra quelli specificati.
meal_category_check	La categoria di un Meal deve essere una tra quelle specificate.
allergen_name_check	Il nome di un allergene deve essere uno tra quelli specificati.
rider_gender_check	Il genere di un Rider deve essere uno tra quelli specificati.
customer_gender_check	Il genere di un Customer deve essere uno tra quelli specificati.

Implementazione dei vincoli di CHECK

```
-- Controlla che gli indirizzi abbiano il seguente formato: "<denominazione urbanistica>, <indirizzo>, <numero civico>, <CAP>, <Comune>, <Provincia>". Non sono case sensitive.
ALTER TABLE Customer
ADD CONSTRAINT customer_address_check CHECK
(address~*'^^(accesso|arco|belvedere|borgo|campo|canale|cascina|cavone|cavalcavia|contrada|corso|cortile|costa|discesa|fondo|galleria|frazione|isola|lido|litoranea|lungo|lungomare|masseria|molo|mura|passaggio|passo|pendio|piazza|piazzale|piazzetta|portico|porto|rampa|recinto|rione|riva|rotonda|salita|scalinata|scesa|sentiero|
spiaggia|spiazzo|strada|stradale|stretto|stretta|strettoia|terrazza|traversa|via|vicoletto|vico|villaggio|viuzza)( )[ A-Za-z0-9]+(, )[ A-Z0-9]+(, )[0-9]{5}(, )[\\x20-\\xA5]+(, )[\\x20-\\xFF]+$' );
```

```
ALTER TABLE CustomerOrder
ADD CONSTRAINT customer_order_address_check CHECK
(address~*'^^(accesso|arco|belvedere|borgo|campo|canale|cascina|cavone|cavalcavia|contrada|corso|cortile|costa|discesa|fondo|galleria|frazione|isola|lido|litoranea|lungo|lungomare|masseria|molo|mura|passaggio|passo|pendio|piazza|piazzale|piazzetta|portico|porto|rampa|recinto|rione|riva|rotonda|salita|scalinata|scesa|sentiero|
spiaggia|spiazzo|strada|stradale|stretto|stretta|strettoia|terrazza|traversa|via|vicoletto|vico|villaggio|viuzza)( )[ A-Za-z0-9]+(, )[ A-Z0-9]+(, )[0-9]{5}(, )[\\x20-\\xA5]+(, )[\\x20-\\xFF]+$' );
```

```
ALTER TABLE Rider
ADD CONSTRAINT rider_address_check CHECK
(address~*'^^(accesso|arco|belvedere|borgo|campo|canale|cascina|cavone|cavalcavia|contrada|corso|cortile|costa|discesa|fondo|galleria|frazione|isola|lido|litoranea|lungo|lungomare|masseria|molo|mura|passaggio|passo|pendio|piazza|piazzale|piazzetta|portico|porto|rampa|recinto|rione|riva|rotonda|salita|scalinata|scesa|sentiero|
spiaggia|spiazzo|strada|stradale|stretto|stretta|strettoia|terrazza|traversa|via|vicoletto|vico|villaggio|viuzza)( )[ A-Za-z0-9]+(, )[ A-Z0-9]+(, )[0-9]{5}(, )[\\x20-\\xA5]+(, )[\\x20-\\xFF]+$' );
```

```
ALTER TABLE Shop
ADD CONSTRAINT shop_address_check CHECK
(address~*'^^(accesso|arco|belvedere|borgo|campo|canale|cascina|cavone|cavalcavia|contrada|corso|cortile|costa|discesa|fondo|galleria|frazione|isola|lido|litoranea|lungo|lungomare|masseria|molo|mura|passaggio|passo|pendio|piazza|piazzale|piazzetta|portico|porto|rampa|recinto|rione|riva|rotonda|salita|scalinata|scesa|sentiero|
spiaggia|spiazzo|strada|stradale|stretto|stretta|strettoia|terrazza|traversa|via
```

```

|vicoletto|vico|villaggio|viuzza)( )[ A-Za-z0-9]+(, )[ A-Z0-9]+(, )[0-
9]{5}(, )[ \x20-\xA5]+(, )[ \x20-\xFF]+$' );
-- Controlla che l' email sia scritta correttamente.
ALTER TABLE Customer
ADD CONSTRAINT customer_email_check CHECK (email ~* '^[A-Za-z0-9._%-]+(@)[A-Za-
z0-9.-]+[.][A-Za-z]+$');

-- Controlla che l' email abbia dominio "foodoverflow.it".
ALTER TABLE Shop
ADD CONSTRAINT shop_email_check CHECK (email ~ '^[A-Za-z0-
9._%-]+(@foodoverflow.it)$');

-- Controlla che il codice fiscale sia scritto correttamente.
ALTER TABLE Customer
ADD CONSTRAINT customer_cf_check CHECK (cf ~ '^[a-zA-Z]{6}[0-
9]{2}[abcdehlmprstABCDEHLMPRST]{1}[0-9]{2}([a-zA-Z]{1}[0-9]{3})[a-zA-Z]{1}$');

-- Controlla che il codice fiscale sia scritto correttamente.
ALTER TABLE Rider
ADD CONSTRAINT rider_cf_check CHECK (cf ~ '^[a-zA-Z]{6}[0-
9]{2}[abcdehlmprstABCDEHLMPRST]{1}[0-9]{2}([a-zA-Z]{1}[0-9]{3})[a-zA-Z]{1}$');

-- Controlla che l' attributo closing_days sia una stringa formata da giorni
della settimana separati da virgola-spazio, senza ripetizioni. Non è case
sensitive.
ALTER TABLE Shop
ADD CONSTRAINT shop_closing_days_check CHECK (closing_days ~*
'^((lunedì|lunedì|martedì|martedì|mercoledì|mercoledì|giovedì|giovedì|venerdì|ve
nerdì|sabato|domenica), (?!$)|$)*$');

-- Controllano che la fasce orarie lavorative siano del formato (HH:MM-HH:MM) .
ALTER TABLE Shop
ADD CONSTRAINT shop_working_hours_check CHECK (working_hours ~* '^(0-2){1}[0-
9]{1})(:[0-5]{1}[0-9]{1})(-)([0-2]{1}[0-9]{1})(:[0-5]{1}[0-9]{1})$');

ALTER TABLE Rider
ADD CONSTRAINT rider_working_hours_check CHECK (working_hours ~* '^(0-2){1}[0-
9]{1})(:[0-5]{1}[0-9]{1})(-)([0-2]{1}[0-9]{1})(:[0-5]{1}[0-9]{1})$');

-- Controllano che il numero di cellulare abbia 10 cifre numeriche.
ALTER TABLE Customer
ADD CONSTRAINT customer_cellphone_check CHECK (cellphone ~ '^[0-9]{10}$');

ALTER TABLE Rider
ADD CONSTRAINT rider_cellphone_check CHECK (cellphone ~ '^[0-9]{10}$');

-- Controlla che il prefisso sia nazionale, seguito da un trattino e da 7 cifre
numeriche.
ALTER TABLE Shop
ADD CONSTRAINT shop_home_phone_check CHECK (home_phone ~
'^((004191|010|011|0121|0122|0123|0124|0125|0131|0141|0142|0143|0144|015|0161|016
3|0165|0166|0171|0172|0173|0174|0175|0182|0183|0184|0185|0187|019|02|030|031|032
1|0322|0323|0324|0331|0332|0341|0342|0343|
0344|0345|0346|035|0362|0363|0364|0365|0371|0372|0373|0374|0375|0376|0377|0381|0
382|0383|0384|0385|0386|039|040|041|0421|0422|0423|0424|0425|0426|0427|0428|0429
|0431|0432|0433|0434|0435|0436|0437|0438|
0439|0442|0444|0445|0445|0461|0462|0463|0464|0465|0471|0472|0473|0474|0481|049|05
0|051|0521|0522|0523|0524|0525|0532|0533|0534|0535|0536|0541|0542|0543|0544|0545
|0546|0547|0549|055|0564|0565|0566|0571|

```

```

0572|0573|0574|0575|0577|0578|0583|0584|0585|0586|0587|0588|059|06|0623|070|071|
0721|0722|0731|0732|0733|0734|0735|0736|0737|0742|0743|0744|0746|075|0761|0763|0
765|0766|0771|0773|0774|0775|0776|0781|
0782|0783|0784|0785|0789|079|080|081|0823|0824|0825|0827|0828|0831|0832|0833|083
5|0836|085|0861|0862|0863|0864|0865|0871|0872|0873|0874|0875|0881|0882|0883|0884
|0885|089|090|091|0921|0922|0923|
0924|0925|0931|0932|0933|0934|0935|0941|0942|095|0961|0962|0963|0964|0965|0966|0
967|0968|0971|0972|0973|0974|0975|0976|0981|0982|0983|0984|0985|099) (-) [0-
9] {7} \$') ;

-- Controlla che un rider sia associato al più a 3 consegne.
ALTER TABLE Rider
ADD CONSTRAINT rider_deliveries_number_check CHECK (deliveries_number >= 0) AND
(deliveries_number<=3);

-- Controlla che il prezzo di un meal sia maggiore di 0.
ALTER TABLE Meal
ADD CONSTRAINT meal_price_check CHECK (price > 0);

-- Controlla che la quantità di un meal nel carrello sia maggiore di 0.
ALTER TABLE OrderComposition
ADD CONSTRAINT order_composition_quantity_check CHECK (quantity > 0);

-- Controlla che lo stato dell'ordine sia uno tra quelli riportati di sotto
nell' espressione regolare. Non è case sensitive.
ALTER TABLE CustomerOrder
ADD CONSTRAINT customer_order_status_check CHECK (status ~* '^ (In attesa|In
consegna|Consegnato|Errore) \$');

-- Controlla che la categoria di un meal sia una tra quelle riportate di sotto
nell' espressione regolare. Non è case sensitive.
ALTER TABLE Meal
ADD CONSTRAINT meal_category_check CHECK (category ~* '^ (Primo
piatto|Carne|Pesce|Pizza|Panino|Fritto|Dolce|Bevanda analcolica|Bevanda
alcolica) \$');

-- Controlla che l' allergene sia uno tra quelli riportati di sotto
nell'espressione regolare. Non è case sensitive.
ALTER TABLE Allergen
ADD CONSTRAINT allergen_name_check CHECK (name ~* '^ (Cereali e
derivati|Crostacei|Uova|Pesce|Arachidi|Soia|Latte e derivati|Frutta a
guscio|Sedano|Senape|Sesamo|An. solforosa e solfiti|Lupini|Molluschi) \$');

-- Controllano che il genere sia "M" oppure "F". Non è case sensitive.
ALTER TABLE Rider
ADD CONSTRAINT rider_gender_check CHECK (gender ~* '^ (M|F) \$');

ALTER TABLE Customer
ADD CONSTRAINT customer_gender_check CHECK (gender ~* '^ (M|F) \$');

```