

DB_KITCHEN

2024

**Progetto di una base di dati per la gestione di una
catena di ristoranti**

Il tuo nome

Gianmaria Di Fronzo

La tua email

gianmaria.difronzo@studio.unibo.it

1. ANALISI

1.1 Analisi dei requisiti

Si vuole sviluppare un gestionale che gestisca il flusso di lavoro in una famosa catena di ristoranti

	<i>Definizione delle specifiche in linguaggio naturale</i>
1	Tenere traccia di tutto il personale in cucina anche chi non lavora più
2	Ogni cuoco deve poter vedere direttamente sul software quali mansioni ha da fare in una specifica data e ora, capendo se sono destinate le preparazioni ad un pranzo o cena
3	Tenere traccia di tutti i cibi e alimenti presenti nella dispensa e nelle celle frigorifero, qualsiasi rifornimento deve essere registrabile e visibile
4	Ogni cibo deve avere tutte le informazioni adeguate per poter essere gestito dai cuochi
5	Ogni cibo utilizzato nelle preparazioni deve essere scalato nelle quantità indicate dalla preparazione che ne fa uso
6	Ogni volta che il cuoco fa la spesa deve poter registrare tutte le nuove scorte di cibo comprate
7	Ogni cuoco deve poter accedere al portale e capire se ci sono abbastanza risorse in dispensa per effettuare la preparazione di un piatto
8	Quando un cibo è ancora presente e supera la data di scadenza il cuoco deve poter capirlo
9	Devono essere visibili tutti i piatti presenti nel menù, ordinabili dai clienti
10	Quando un cliente ordina un piatto il cameriere lo deve aggiungere all'ordine relativo al specifico numero di tavolo
11	Alla fine del pasto un cameriere deve far pagare il conto al cliente sulla somma di tutti i pasti e bevande che ha ordinato

12	La cucina deve visualizzare durante il servizio gli ordini registrati dai camerieri
13	Se un cliente esegue una richiesta extra non presente nel menù il cameriere deve poter comunque aggiungere il prezzo sul conto
14	Se un cliente chiede un piatto extra non presente nel menù bisogna comunque tenere traccia degli alimenti usati nel magazzino o cella frigorifero
15	Sul menù devono essere indicati tutti gli allergeni appartenenti ad un piatto indicato
16	Le scorte non dovrebbero solo aggiornarsi ma comunque bisogna tenere traccia di tutte le scorte usate per mesi
17	Ogni tavolo ha un totale di posti e quando i clienti arrivano il cameriere apre un conto inizialmente vuoto sul tavolo
18	Durante il pasto vengono aggiunti ordini sullo scontrino del tavolo
19	Quando il tavolo deve essere liberato il cameriere deve vedere il tavolo come libero
20	Il cameriere deve poter far pagare meno gli amici dello chef e i tavoli più numerosi
21	Il personale deve poter visualizzare i top 3 piatti ordinati nel menù
22	Il personale deve poter visualizzare la media dell'incasso mensile
23	Il personale del reparto cucina deve poter visualizzare i 3 cuochi con meno mansioni assegnate

A seguito della lettura e comprensione dei requisiti richiesti dal cliente, si procede sviluppando un testo che ne riassume tutti i concetti e in particolare ne estragga quelli principali, risultando essere in questo modo meglio fruibile per la realizzazione della base di dati. Si tiene conto delle seguenti correzioni di ambiguità.

Riga	Termine	Nuovo Termine
1	Personale in cucina	Cuoco
2	Pranzo o cena	Pasto
3-14	Alimenti nella cella frigorifero o dispensa	Dispensa
4-5-6	Cibo	Alimento
7-15	Piatto	Portata

18	Scontrino	Conto
20	Sconti amici e tavoli numerosi	Sconto

	<i>Rilevamento ambiguità e correzioni proposte</i>
1	Tenere traccia di tutti i cuochi , anche chi non lavora più
2	Ogni cuoco deve poter vedere direttamente sul software quali mansioni ha da fare in una specifica data e ora, capendo se sono destinate le preparazioni ad un pasto
3	Tenere traccia di tutta la dispensa , qualsiasi rifornimento deve essere registrabile e visibile
4	Ogni alimento deve avere tutte le informazioni adeguate per poter essere gestito dai cuochi
5	Ogni alimento utilizzato nelle preparazioni deve essere scalato nelle quantità indicate dalla preparazione che ne fa uso
6	Ogni volta che il cuoco fa la spesa deve poter registrare tutte le nuove scorte di alimenti comprati
7	Ogni cuoco deve poter accedere al portale e capire se ci sono abbastanza risorse in dispensa per effettuare la preparazione di un portata
8	Quando un cibo è ancora presente e supera la data di scadenza il cuoco deve avere capirlo
9	Devono essere visibili tutti i piatti presenti nel menù, ordinabili dai clienti
10	Quando un cliente ordina un portata il cameriere lo deve aggiungere all'ordine relativo al specifico numero di tavolo
11	Alla fine del pasto un cameriere deve far pagare il conto al cliente sulla somma di tutti i pasti e bevande che ha ordinato
12	La cucina deve visualizzare durante il servizio gli ordini registrati dai camerieri
13	Se un cliente esegue una richiesta extra non presente nel menù il cameriere deve poter comunque aggiungere il prezzo sul conto
14	Se un cliente chiede una portata extra non presente nel menù bisogna comunque tenere traccia delle scorte nella dispensa
15	Sul menù devono essere indicati tutti gli allergeni appartenenti ad una portata

	indicata
16	Le scorte non dovrebbero solo aggiornarsi ma comunque bisogna tenere traccia di tutte le scorte usate per mesi
17	Ogni tavolo ha un totale di posti e quando i clienti arrivano il cameriere apre un conto inizialmente vuoto sul tavolo
18	Durante il pasto vengono aggiunti ordini sullo conto del tavolo
19	Quando il tavolo deve essere liberato il cameriere deve vedere il tavolo come libero
20	Il cameriere deve poter fare uno sconto a gli amici dello chef e ad i tavoli più numerosi
21	Il personale deve poter visualizzare i top 3 piatti ordinati nel menù
22	Il personale deve poter visualizzare la media dell'incasso mensile
23	Il personale del reparto cucina deve poter visualizzare i 3 cuochi con meno mansioni assegnate

1.2 Progetto viste

Nel seguito vengono considerate le due diverse aree di utenza del DB:

Cucina

Salone

1.2.1 Formulazione e analisi requisiti cucina

In ogni cucina di un ristorante vi è un team di cuochi che durante gli orari di lavoro effettuano delle preparazioni in vista del prossimo pasto. I cuochi possono dividersi in sottocategorie ognuna con ruolo preciso, le categorie sono le seguenti:

- *Direttore cucina*
- *Capo cuoco*
- *Sous chef*
- *Capo partita primi*
- *Capo partita secondi*
- *Demi chef de partie*
- *Aiuto cuoco*

Il direttore cucina registra tutte le mansioni per ogni specifico cuoco, in modo che ognuno possa visionare esattamente cosa deve fare

Ogni qual volta che una mansione viene portata a termine essa corrisponde esattamente a delle quantità di alimenti presenti nella dispensa che verranno scalate. In questo modo sarà possibile visionare l'esaurimento scorte.

Gli alimenti presenti nella dispensa si possono suddividere nelle seguenti categorie:

- *Carne*
- *Pesce*
- *Verdure*
- *Legumi*
- *Frutta*
- *Spezie*
- *Latticini*
- *Bevande*
- *Altro*

Elenco delle principali azioni dei cuochi

1	Visualizzare per ogni cuoco le mansioni nella giornata corrente o in una data indicata, con relativo orario e relativi alimenti necessari alla preparazione
2	Inserire alimenti nella dispensa con relativo nome, data di scadenza, quantità espressa in kg
3	Visionare tutti gli alimenti presenti nella dispensa con relative informazioni per ogni alimento
4	Modificare delle preparazioni all'occorrenza
5	inserire le preparazioni assegnando ad un specifico cuoco il relativo orario, data, alimenti necessari, scalando le quantità inserite per ogni prodotto della dispensa
6	Leggere facilmente gli alimenti vicini alla data di scadenza.
7	Modificare il manualmente le quantità di un prodotto presente nella dispensa
8	Visualizzare quali cuochi passati hanno lavorato nella cucina
9	Rinnovare il contratto ad un cuoco

10	Aggiungere il contratto di un nuovo cuoco
11	Alzare il grado di un cuoco e quindi modificarne il ruolo in cucina

1.2.2 Formulazione e analisi requisiti salone

In ogni ristorante registrato vi è un salone addebito al servizio al tavolo, in quest'area il personale dei camerieri deve tenere traccia dei tavoli presenti, quante persone sono sedute ad un singolo tavolo, quali sono occupati e per ognuno se occupato aprire un conto in sospeso che verrà successivamente chiuso al momento del saldo.

Durante il servizio i camerieri devono poter inviare gli ordini al reparto cucina, è superfluo però tenere traccia delle ordinazioni effettuate all'infuori dell'attuale servizio.

Su ogni ordine deve esserci traccia dei piatti ordinati e da quale cameriere è stato registrato e servito. Inoltre l'ordine è destinato ad un tavolo e del proprio scontrino finale è sufficiente salvare esclusivamente il totale pagato in modo tale da averne uno storico sul quale contare gli incassi a fine mese.

Il conto deve tenere salvato ogni pasto consumato, le bevande, ed in caso gli alimenti extra ordinati singolarmente dal tavolo, ognuno di questi elementi avrà un costo che verrà riportato poi sullo scontrino insieme al conto totale da dover saldare.

1	Registrare un nuovo tavolo
2	Aggiungere un ordine ad un tavolo
3	Eliminare l'ordine di un tavolo
4	Calcolare il totale sullo scontrino di un tavolo
5	Chiudere un conto e rendere libero il tavolo
6	Unire gli ordini di due tavoli
7	Rinnovare il contratto ad un cameriere
8	Registrare un nuovo cameriere
9	Inviare l'ordine di un tavolo direttamente al reparto cucina
10	Gestire in maniera efficiente l'interazione durante il servizio cucina-salone

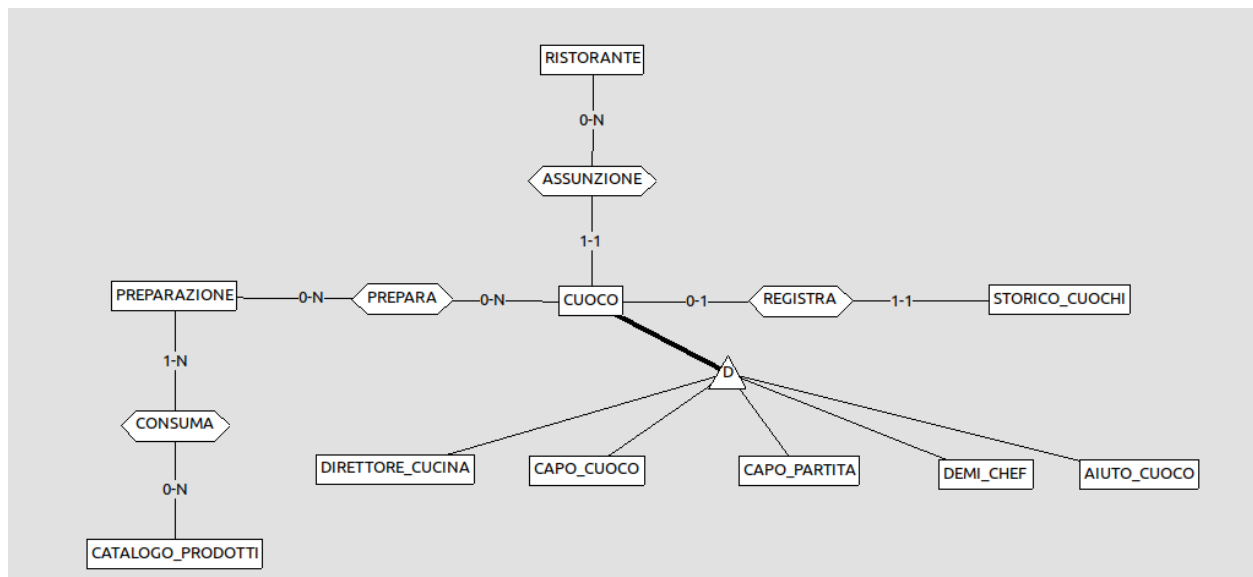
attraverso le ordinazioni

2. PROGETTAZIONE CONCETTUALE

2.1 CUCINA

2.1.1 Progettazione dello schema E/R

Dopo aver esaminato il dominio del problema e le richieste da parte del reparto cucina, viene proposto il seguente schema scheletro:



2.1.2 Raffinamenti proposti

L'entità **CUOCO** è raggruppabile in un'entità padre più generica e flessibile **DIPENDENTE**, per questo motivo l'entità cuoco sarà una gerarchia di DIPENDENTE.

L'entità **PREPARAZIONE** fa molto spesso riferimento ad una consultazione giornaliera, le preparazioni già eseguite, soprattutto quelle riferite ai giorni passati, sono consultate di rado dai dipendenti, perciò si decide di dividere l'entità in due categorie con associazione fra di esse: **PREPARAZIONE** e **STORICO_PREPARAZIONI**.

L'entità **CATALOGO_PRODOTTI** potrebbe creare delle divergenze inserendo alimenti manualmente, si decide quindi di inserire un'entità esterna **CATALOGO_PRODOTTI**, il quale racchiude un mero elenco di tutti gli alimenti esistenti sul mercato italiano (eventualmente importati). Tale entità si collega tramite associazione a **DISPENSA** la quale rappresenta tutti gli alimenti presenti nelle riserve della cucina, o meglio definita dispensa.

La nuova entità **DISPENSA** sarà identificata dalla data di arrivo e dall'alimento.

Tutti le preparazioni sono destinate alla creazione di un pasto sul menù, per questo motivo si decide di integrare un'entità **MENU** legata con associazione one-to-many all'entità **PREPARAZIONE**.

Ogni portata del menù è composta da una ricetta che richiede un totale di porzioni di alimenti, al fine di ciò viene inserita una associazione **RICETTARIO** che collega ogni portata del menù con un insieme finito di elementi.

Attenzione: per ogni portata nel menù non è detto vi sia una preparazione minima maggiore di zero in quanto una portata potrebbe essere inserita nel menù per la clientela, ma senza essere mai ordinata o deciso di preparare.

Ogni cuoco può avere un particolare ruolo nella cucina, perciò viene integrato ogni ruolo come gerarchia di **CUOCO**, le entità integrate sono le seguenti: **DIRETTORE_CUCINA**, **CAPO_CUOCO**, **SOUS_CHEF**, **CAPO_PARTITA_PRIMI**, **CAPO_PARTITA_SECONDI**, **DEMI_CHEF**, **AIUTO_CUOCO**.



Quello che era lo storico cuochi, data la generalizzazione a **DIPENDENTE**, ora diventa un'entità **STORICO_DIPENDENTI**.

Data questa modifica il campo data di fine contratto non sarà necessario sull'entità **DIPENDENTE** ma solo in **STORICO_DIPENDENTI**.

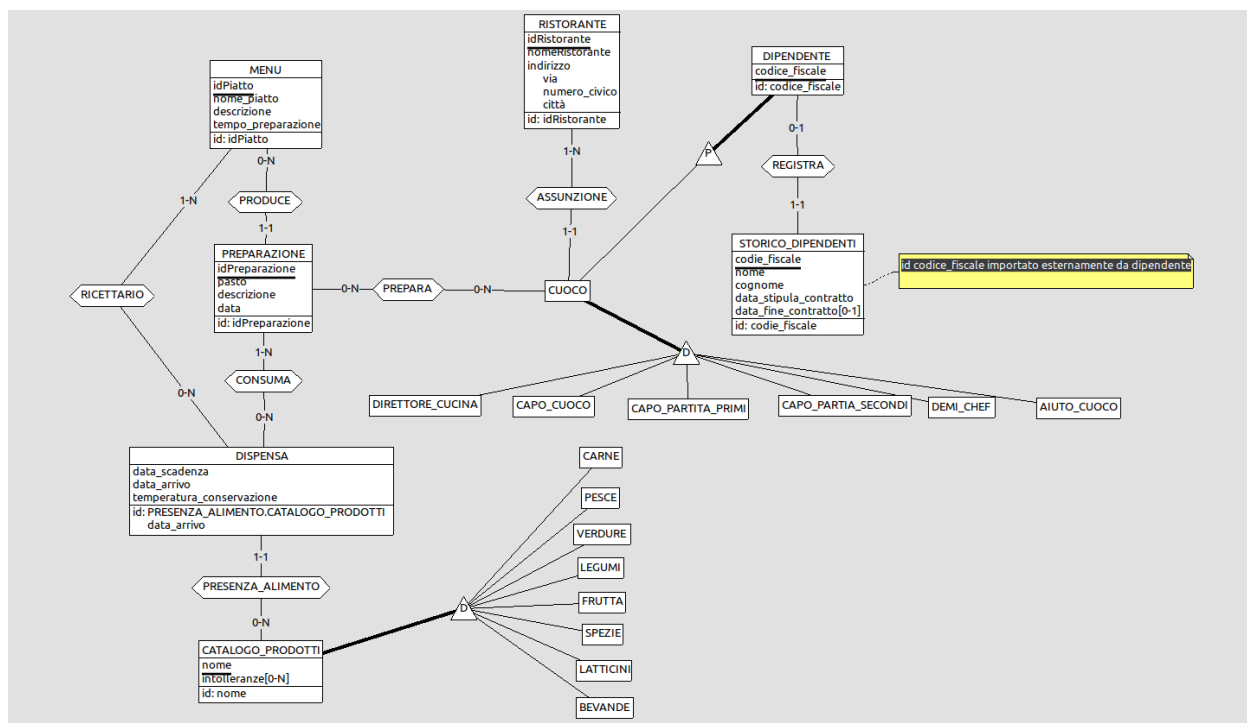
Infatti al momento dell'inserimento di un dipendente verrà automaticamente inserito nello storico, al momento del termine contratto il dipendente deve essere eliminato dall'entità **DIPENDENTE** e aggiornato nell'entità **STORICO_DIPENDENTI**.

Ogni preparazione è destinata ad un pasto della giornata (colazione, pranzo, cena) perciò si intraprende la soluzione di aggiungere un campo a **PREPARAZIONE**: *pasto*. Nella valutazione si è tenuto in conto l'idea di creare un'entità PASTO, non intrapresa in quanto i pasti nella giornata sono statici e non variabili.

Le tipologie degli alimenti sono multiple ma non variabili, rinunciando alla flessibilità si intraprende una risoluzione che avvantaggia per semplicità e chiarificazione dei concetti implementando una gerarchia di tipologie di alimento. Le tipologie inserite come identità sono le seguenti: **CARNE, PESCE, VERDURE, LEGUMI, FRUTTA, SPEZIE, LATTICINI, BEVANDE**.

2.1.3 Schema concettuale parziale

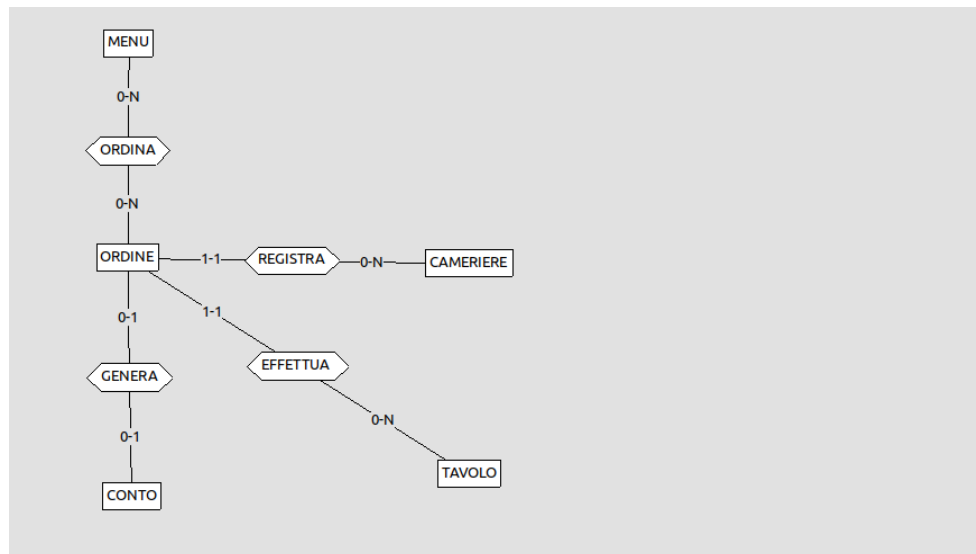
Dopo l'applicazione dei raffinamenti proposti e l'inserimento degli attributi necessari per le varie entità, lo schema E/R finale per la cucina risulta essere di questo tipo:



2.2 SALONE

2.2.1 Progettazione dello schema E/R

Dopo aver esaminato il dominio del problema e le richieste da parte del reparto salone, viene proposto il seguente schema scheletro:



2.2.2 Raffinamenti proposti

L'entità **CAMERIERE** è raggruppabile in un'entità padre più generica e flessibile **DIPENDENTE**, per questo motivo l'entità cameriere sarà una gerarchia di **DIPENDENTE**.

Il totale di un conto deve essere calcolato sul totale dei prezzi di ogni portata inserita dentro l'ordine direttamente dal menù, a tale proposito si inserisce una relazione ordine **GENERA** conto che associa ogni ordine a min=0, max=1 conti. Ogni conto ha min=0, max=N ordini associati.

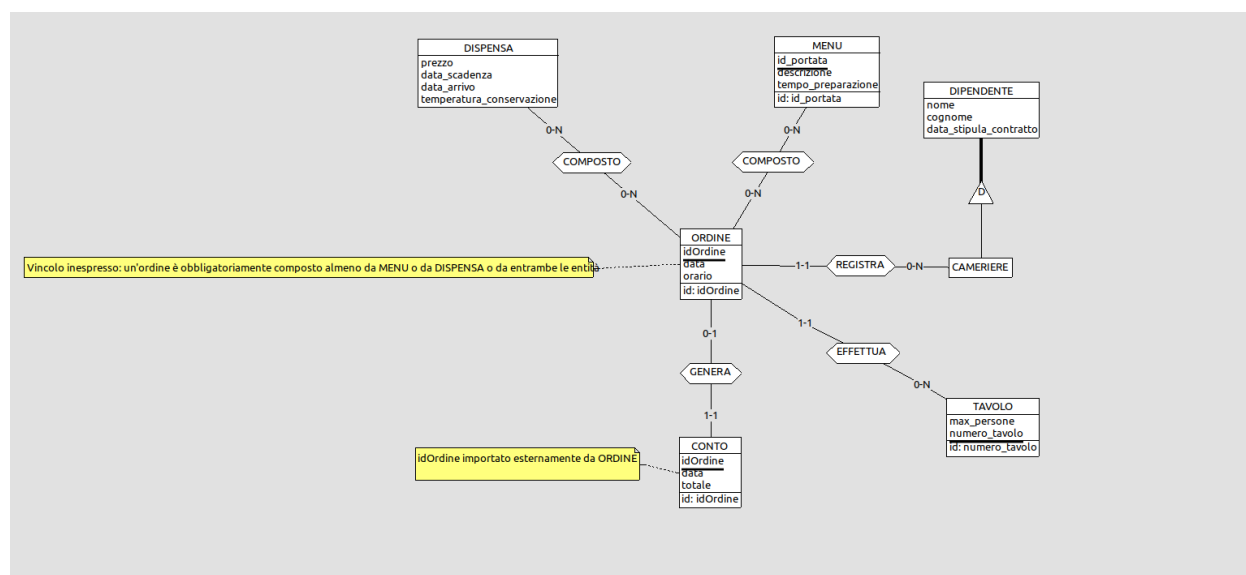
Un ordine può essere composto da un alimento (bevande, salumi, ecc) o da portate del menù, perciò vengono inserite le relazioni "**composto**" che associano gli ordini o all'entità **MENU** o a **DISPENZA**, caratterizzate da un vincolo inespresso che comunica che ogni ordine deve corrispondere obbligatoriamente almeno all'entità **MENU** o **DISPENZA** o entrambe. Si presenta però una problematica, sullo schema ER proposto della CUCINA (punto 2.1) ogni alimento presente nell'entità **DISPENZA** non ha associato un campo valore economico da sommare direttamente al totale del conto, questo problema viene decifrato attraverso l'inserimento di un campo *prezzo* nell'entità **DISPENZA**

Ogni ordine ha un id composto dato da *idOrdine*, **TAVOLO**, **CAMERIERE**

Si tiene presente che ogni tavolo è presente in database perennemente fino a scelta contraria degli utenti, per questo motivo è sufficiente registrare se un tavolo è libero o meno. A tal proposito l'associazione Tavolo ordine viene mantenuta con cardinalità Tavolo-Ordine 0-N.

2.2.3 Schema concettuale parziale

Dopo l'applicazione dei raffinamenti proposti e l'inserimento degli attributi necessari per le varie entità, lo schema E/R finale per la cucina risulta essere di questo tipo:



2.4 INTEGRAZIONE DELLE VISTE

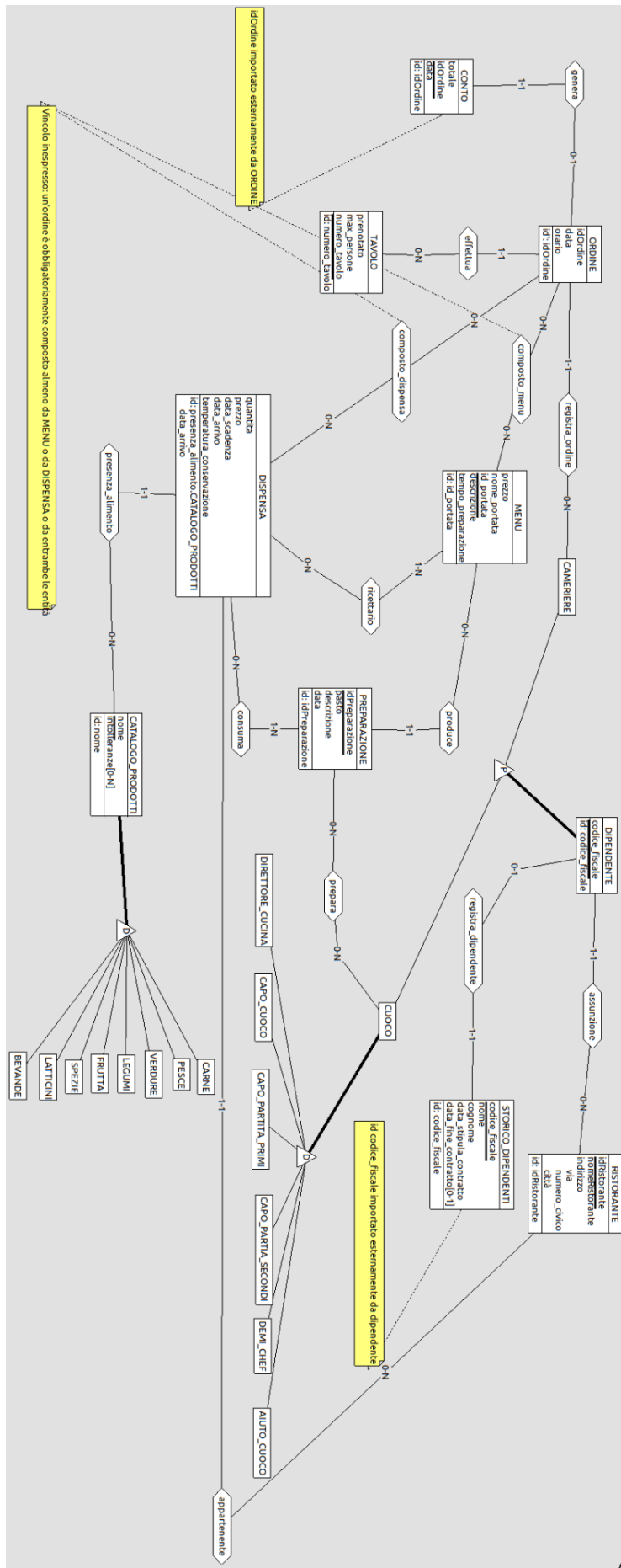
Le due viste, che compaiono all'interno della base di dati, hanno elementi in comune facilmente individuabili osservando gli schemi E/R finali proposti per il reparto cucina e per il reparto salone. Nello specifico i punti di giunzione sono il **DISPENSA**, **MENU** e **DIPENDENTE**. Non esistono altri elementi in comune tra le due viste per cui non si evidenzia alcun problema associato a possibili conflitti di nomi.

Nello schema concettuale l'entità **DISPENSA** non aveva un riferimento diretto al ristorante dove risiede. Per questo motivo è stata aggiunta la relazione uno a molti "**appartenente**" tra **DISPENSA** e **RISTORANTE**.

Inoltre viene inserito il campo prezzo sull'entità **MENU** per la scrittura dell'entità **CONTO**, la quale possiede il campo "totale" che rappresenta il saldo del conto dato dal prezzo degli ordini fatti.

2.5 SCHEMA CONCETTUALE

La pagina successiva è dedicata al diagramma E/R completo.



3. PROGETTAZIONE LOGICA

3.1 Stima del volume dei dati

Si fornisce in questa fase una tabella contenente il numero medio di istanze di ogni entità e associazione dello schema globale: i valori di carico fanno riferimento alle partecipazioni medie riportate nell'immagine dello schema E/R al paragrafo 2.5

Concetto	Tipo (Entity/Relationship)	Volume
RISTORANTE	E	20
DIPENDENTE	E	2000
assunzione	R	2000
STORICO_DIPENDENTI	E	3000
registra_dipendente	R	2000
CUOCO	E	1300
CAMERIERE	E	700
DIRETTORE_CUCINA	E	20
CAPO_CUOCO	E	130
CAPO_PARTITA_PRIMI	E	130
CAPO_PARTITA_SECONDI	E	130
DEMI_CHEF	E	130
AIUTO_CUOCO	E	160
registra_ordine	R	20000
ORDINE	E	20000

MENU	E	400
TAVOLO	E	500
composto_menu	R	13000
composto_dispensa	R	7000
effettua	R	20000
DISPENSA	E	6000
CATALOGO_PRODOTTI	E	3000
CARNE	E	200
PESCE	E	200
VERDURE	E	100
LEGUMI	E	100
FRUTTA	E	50
SPEZIE	E	60
LATTICINI	E	50
BEVANDE	E	40
presenza_alimento	R	300
ricettario	R	500
PREPARAZIONE	E	40000
prepara	E	50000
genera	R	20000
CONTO	E	20000
consuma	R	50000
appartenente	R	6000

3.2 Descrizione delle operazioni principali e stima della loro frequenza

In questa fase si propone una tavola delle operazioni utilizzata per costituire una stima delle principali operazioni richieste dal settore CUCINA e dal settore SALONE. Per la descrizione dettagliata di tali operazioni si vedano i paragrafi: 1.2.1 e 1.2.2.

Cod. operazione	Nome operazione	Frequenza
O1	inserimento ordine	400 al giorno
O2	visualizzare uno scontrino in una data specifica	100 alla settimana
O3	generare scontrino	400 al giorno
O4	prenotare un tavolo	200 al giorno
O5	assegnare preparazioni ai cuochi	600 al giorno
O6	inserimento di nuovi prodotti nella dispensa	100 alla settimana
O7	modificare le quantità rimanenti di un alimento nella dispensa	100 al giorno
O8	visualizzare le mansioni di un cuoco nella giornata	400 al giorno
O9	rinnovare il contratto ad un dipendente	2000 all'anno
O10	modificare uno o più piatti nel menù	300 al mese
O11	aumentare di grado un cuoco	300 all'anno
O12	assumere un nuovo dipendente	100 all'anno
O13	modificare il prezzo di vendita di un alimento nella dispensa	100 all'anno
O14	ricercare una preparazione fatta in una data specifica	100 al giorno

O15	visualizzare il fatturato totale annuo	1 all'anno
O16	aggiungere coperti ad un specifico tavolo	80 al giorno
O17	ricercare un vecchio dipendente nello storico dipendenti	20 alla settimana
O18	Visualizzare i top 3 piatti ordinati dal menù	1 all'anno
O19	Visualizzare la media dell'incasso mensile	1 al mese
O20	Visualizzare i primi 4 cuochi con meno preparazioni assegnate	1 al mese

3.3 Schemi di navigazione e tabelle degli accessi

Dopo aver determinato il volume dei dati ed aver associato a ciascuna operazione principale richiesta la propria frequenza di esecuzione, si procede determinando lo schema di navigazione di riferimento per le principali operazioni richieste e si associa ad ognuna di essa anche la relativa tavola degli accessi. Le operazioni per cui questo passaggio risulta essere banale o per cui risulta essere equivalente a quello di una operazione già vista, non verranno considerate. Nel calcolo degli accessi si stima come doppio il peso degli accessi in scrittura, rispetto a quelli in lettura.

01) inserimento ordine:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
ORDINE	E	1	S
TAVOLO	E	1	L
DISPENSA	E	$7000/6000=1$	L
composto_dispenza	A	1	S

MENU	E	$13000/400=32$	L
composto_menu	A	32	S

TOT: $6 \times 600 = 3600$ al giorno

03) generare scontrino:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CONTO	E	1	S
ORDINE	E	1	L
composto_menu	A	$13000/400=32$	L
MENU	E	32	L
composto_dispensa	A	$7000/6000=1$	L
DISPENSA	E	1	L

TOT: $2+1+32+32+1+1 = 69 \times 400 = 27600$ al giorno

05) assegnare preparazioni ai cuochi:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
PREPARAZIONE	E	1	S
consuma	R	$50000/40000=1$	S
DISPENSA	E	$50000/40000=1$	L
MENU	L	1	L

TOT: $6 \times 600 = 3600$ al giorno

06) inserimento nuovi prodotti nella dispensa:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
DISPENSA	E	1	S
CATALOGO_PRODOTTI	E	1	L

TOT: $3 \times 100 = 300$ a settimana

07) modificare le quantità rimanenti di un alimento nella dispensa:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
DISPENSA	E	1	L
DISPENSA	E	1	S
CATALOGO_PRODOTTI	E	1	L

TOT: $4 \times 100 = 400$ al giorno

08) visualizzare le mansioni di un cuoco nella giornata:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
PREPARAZIONE	E	$50000/1300=38$	L
prepara	E	38	L
MENU	E	1	L
DISPENSA	E	$50000/40000=1$	L
consuma	E	$50000/40000=1$	L

TOT: $38+38+1+1+1 = 79 \times 400 = 31600$ al giorno

012) assumere un nuovo dipendente:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
DIPENDENTE	E	1	S
RISTORANTE	E	1	L
STORICO_DIPENDENTI	E	1	S

TOT: $5 \times 100 = 500$ all'anno

014) ricercare una preparazione fatta in una specifica data:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
PREPARAZIONE	E	1	L
consuma	R	$50000/40000=1$	S
DISPENSA	E	$50000/40000=1$	L
MENU	L	1	L

TOT: $5 \times 100 = 500$ al giorno

018) visualizzare i top 3 piatti ordinati dal menù:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
MENU	E	400	L
composto_menù	R	$400 \cdot (130000/200000)$ = 260	L
ORDINE	E	260	L

TOT: $400 + 260 + 260 = 920$ all'anno

019) media incasso mensile:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CONTO	L	20000	L

TOT: 20000 al mese

020) Visualizzare i primi 4 cuochi con meno preparazioni assegnate:

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CUOCO	E	1300	L
prepara	R	$1300 \cdot (500000/400000)$ = 1625	L
PREPARAZIONE	E	1625	L

TOT: $13000 + 1625 + 1625 = 4550$ al mese

3.4 Analisi delle ridondanze

Nelle operazioni 2 e 3 al paragrafo 3.3 vi è la presenza di un campo ridondante. Tale campo è "totale" nell'entità CONTO. Di seguito si pone in analisi la ridondanza e si valuta sulle operazioni 2 e 3 se conviene mantenerla o eliminarla.

ANALISI CON RIDONDANZA:

OPERAZIONE 2: visualizzare uno scontrino in una data specifica.

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CONTO	E	1	L

TOT: $1 \times 100 = 100$ alla settimana

OPERAZIONE 3: generare scontrino

Per eseguire un accesso in scrittura sull'entità **CONTO** è necessario prima calcolare il campo "totale" dato dal prezzo di tutti i record dell'entità **MENU** in associazione "composto_menu" con ORDINE e dal prezzo di tutti i record dell'entità **DISPENSA** in associazione "composto_dispensa" con ORDINE.

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CONTO	E	1	S
ORDINE	E	1	L
composto_menu	A	$13000/400=32$	L
MENU	E	32	L
composto_dispensa	A	$7000/6000=1$	L
DISPENSA	E	1	L

TOT: $2+1+32+32+1+1 = 69 \times 400 = 27600$ al giorno

Il totale degli accessi con ridondanza valutata sulle operazioni risulta essere:

$100 + (27600 \times 7) = 193.300$ a settimana

ANALISI SENZA RIDONDANZA:

OPERAZIONE 2: visualizzare uno scontrino in una data specifica.

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CONTO	E	1	L
ORDINE	E	1	L
composto_menu	A	$13000/400=32$	L
MENU	E	32	L
composto_dispensa	A	$7000/6000=1$	L
DISPENSA	E	1	L

TOT: $1+1+32+32+1+1 = 68 \times 100 = 6800$ alla settimana

OPERAZIONE 3: generare scontrino

In questo caso con assenza di ridondanza non è necessario calcolare il totale del conto durante l'inserimento.

Concetto	Costrutto	Accessi	Tipo (lettura/scrittura)
CONTO	E	1	S
ORDINE	E	1	L

TOT: $2 + 1 = 3 \times 400 = 1200$ al giorno

Il totale degli accessi senza ridondanza valutata sulle operazioni risulta essere:

$6800 + (1200 \times 7) = 15.200$ a settimana

Conclusione

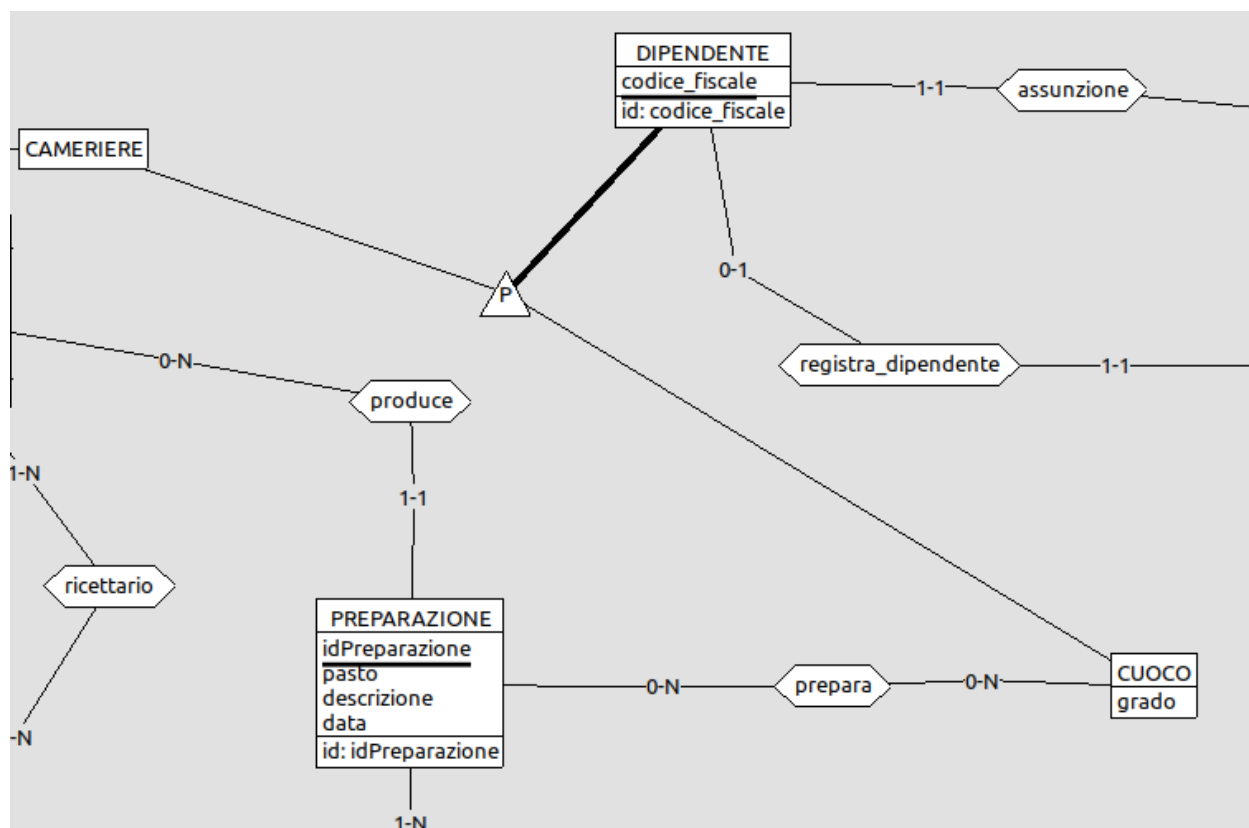
Dalle analisi svolte sulla ridondanza, emerge chiaramente la convenienza a togliere la ridondanza, per questo motivo l'entità conto verrà aggiornata eliminando il campo "totale".

3.5 Raffinamento dello schema

Eliminazione delle gerarchie

Vi sono tre gerarchie da eliminare in totale:

Per la gerarchia **CUOCO** (p,e) in quanto le entità figlie non hanno attributi o associazioni con altre entità, si decide di effettuare un collasso verso l'alto, aggiungendo il campo "grado" all'entità cuoco.



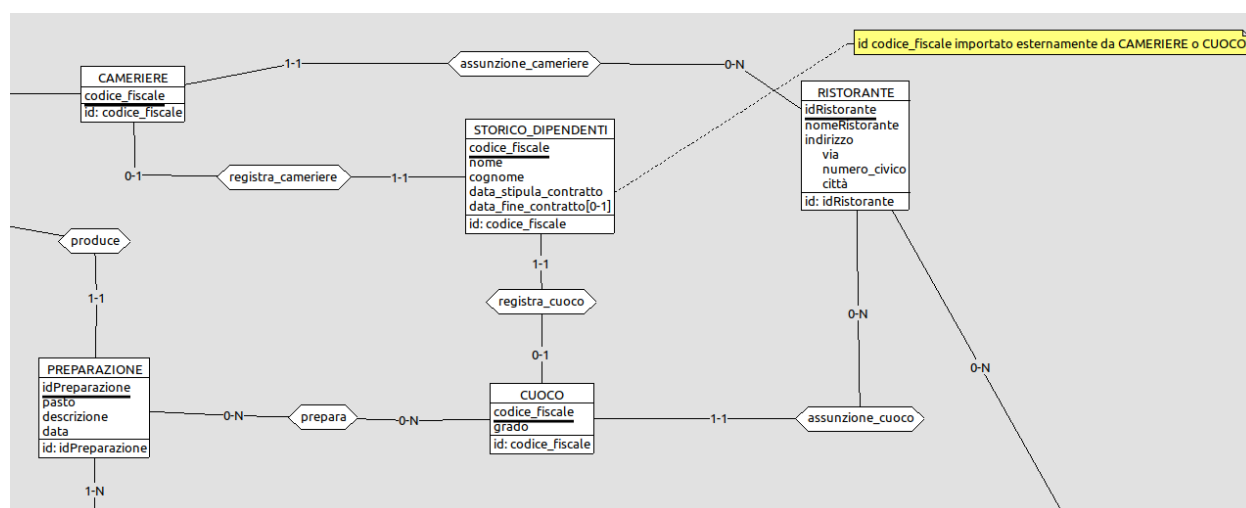
Per la gerarchia **DIPENDENTE** la complessità nella scelta risulta maggiore in quanto le sotto entità **CAMERIERE** e **CUOCO** hanno associazioni uniche, infatti il cameriere non può assegnare preparazioni, il cuoco non può effettuare ordini ai tavoli.

A questo punto restano due opzioni: effettuare un collasso verso il basso oppure mantenere le entità e sostituire la gerarchia con delle associazioni.

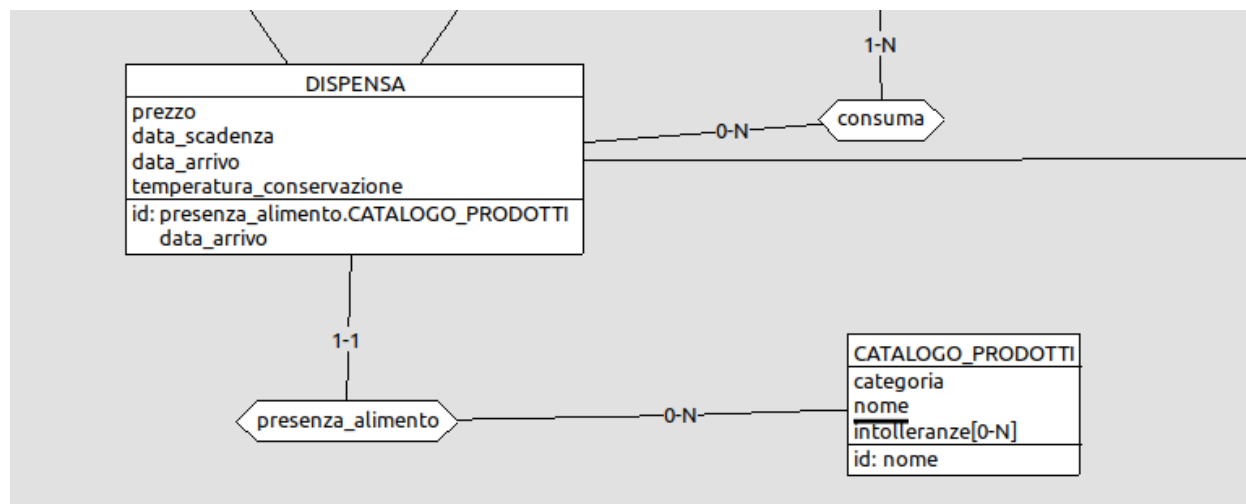
Bisogna però tenere in considerazione l'entità **STORICO_DIPENDENTI** che importa esternamente l'id codice fiscale dall'entità DIPENDENTE.

Si decide di attuare un collasso verso il basso incorporando il campo docide fiscale come id nell'entità **CUOCO** e nell'entità **CAMERIERE**. Si inseriscono quindi le seguenti associazioni:

assunzione cameriere, registra cameriere, assunzione cuoco, registra cuoco, si inserisce inoltre il seguente vincolo inespresso sull'entità **STORICO_DIPENDENTI**:
id codice_fiscale importato esternamente da **CAMERIERE** o **CUOCO**



Per la gerarchia **CATALOGO_PRODOTTI** (p,e) si effettua un collasso verso l'alto, aggiungendo il campo "categorie" per le stesse motivazioni della gerarchia cuoco.



Eliminazione degli attributi composti

Gli unici attributi composti che sono presenti nello schema risultano essere due:

“indirizzo” nell’entità RISTORANTE, si decide quindi di scorporare indirizzo in tre campi: via, numero_civico, citta.

L’altro è un attributo multiplo intolleranze che viene tradotto di un’entità a parte INTOLLERANZA con associazione “intollerante”.

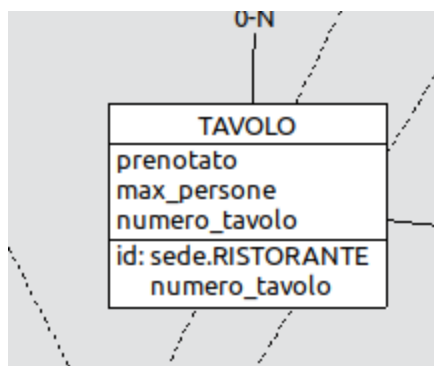
Scelta delle chiavi primarie

Nello schema E/R proposto sono già evidenziate tutte le chiave primarie.

L’unica chiave che viene cambiata è quella di DISPENSA, in quanto andrebbe a complicare notevolmente la relazione “composto_dispensa”

Viene deciso anche di aggiungere la relazione “sede” all’entità tavolo in quanto il “numero tavolo” non bastava per rendere univoco l’id dell’entità nel database.

Le seguenti modifiche:



3.6 Traduzione delle entità e associazioni in relazioni

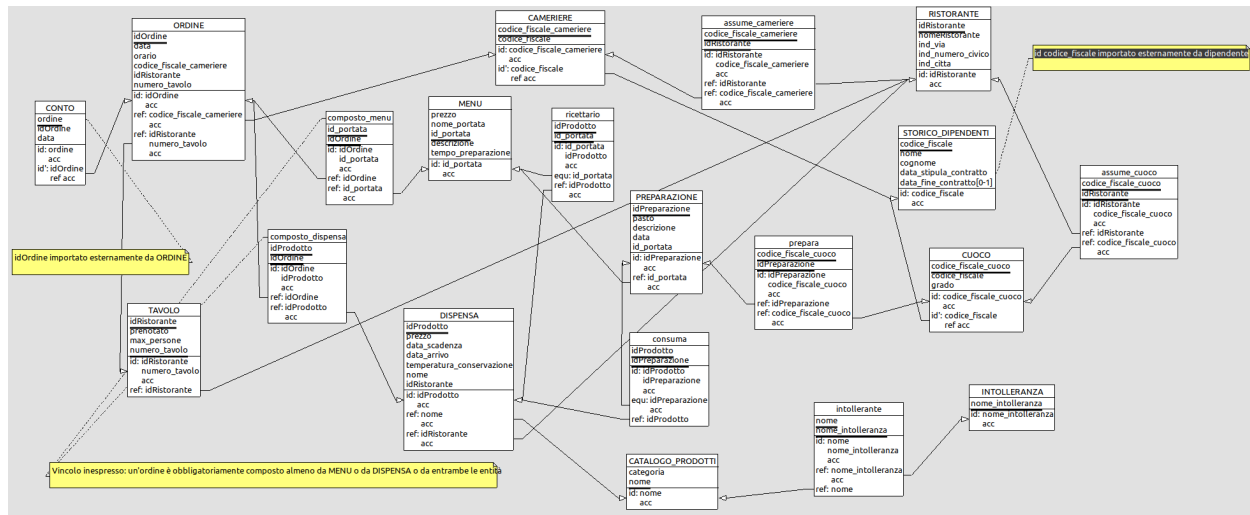
assume_cameriere (codice_fiscale: CAMERIERI, idRistorante: RISTORANTI)

assume_cuoco (codice_fiscale: CUOCHI, idRistorante: RISTORANTI)

CAMERIERI (codice_fiscale)

CATALOGO_PRODOTTI (categoria,nome);
composto_dispensa (ID_ORD: ORDINI, idProdotto: DISPENSA);
composto_menu (ID_ORD: ORDINI, id_portata: MENU);
consuma (idProdotto: DISPENSA, idPreparazione: PREPARAZIONI);
CONTI (ordine: ORDINI, data);
CUOCHI (codice_fiscale, grado);
DISPENSA (idProdotto, prezzo, data_scadenza, data_arrivo, temperatura_conservazione, nome: CATALOGO_PRODOTTI, idRistorante: RISTORANTI);
intollerante (nome_intolleranza: INTOLLERANZE, nome: CATALOGO_PRODOTTI);
INTOLLERANZE (nome);
MENU (prezzo, nome_portata, id_portata, descrizione, tempo_preparazione);
ORDINI (idOrdine, data, orario, codice_fiscale: CAMERIERI, numero_tavolo: TAVOLI);
prepara (codice_fiscale: CUOCHI, idPreparazione: PREPARAZIONI);
PREPARAZIONI (idPreparazione, pasto, descrizione, data, id_portata: MENU);
ricettario (idProdotto: DISPENSA, id_portata: MENU);
RISTORANTE (idRistorante, nomeRistorante, ind_via, ind_numero_civico, ind_citta);
STORICO_DIPENDENTI (codice_fiscale: CUOCO oppure CAMERIERE, nome, cognome, data_stipula_contratto, data_fine_contratto, codice_fiscale_cuoco*, codice_fiscale_cameriere*);
TAVOLO (numero_tavolo, idRistorante: RISTORANTI, prenotato, max_persone);

3.7 Schema relazionale finale



3.8 Costruzione delle tabelle del DB in linguaggio SQL

create table assume_cameriere (

 codice_fiscale_cameriere char(16) not null,

 idRistorante int not null,

 constraint ID_assume_cameriere_ID primary key (idRistorante,codice_fiscale_cameriere));

create table assume_cuoco (

 codice_fiscale_cuoco char(16) not null,

 idRistorante int not null,

 constraint ID_assume_cuoco_ID primary key (idRistorante, codice_fiscale_cuoco));

create table CAMERIERE (

 codice_fiscale_cameriere char(16) not null,

 codice_fiscale char(40) not null,

 constraint ID_CAMERIERE_ID primary key (codice_fiscale_cameriere),

 constraint FKregistra_cameriere_ID unique (codice_fiscale));

```
create table CATALOGO_PRODOTTI (  
    categoria char(40) not null,  
    nome char(40) not null,  
    constraint ID_CATALOGO_PRODOTTI_ID primary key (nome));
```

```
create table composto_dispensa (  
    idProdotto int not null,  
    idOrdine int not null,  
    constraint ID_composto_dispensa_ID primary key (idOrdine, idProdotto));
```

```
create table composto_menu (  
    id_portata int not null,  
    idOrdine int not null,  
    constraint ID_composto_menu_ID primary key (idOrdine, id_portata));
```

```
create table consuma (  
    idProdotto int not null,  
    idPreparazione int not null,  
    constraint ID_consumo_ID primary key (idProdotto, idPreparazione));
```


```
create table CONTO (  
    totale float(1) not null,  
    ordine int not null,  
    idOrdine int not null,  
    data date not null,  
    constraint ID_CONTO_ID primary key (ordine),  
    constraint FKgenera_ID unique (idOrdine));
```

```
create table CUOCO (  
    codice_fiscale_cuoco char(16) not null,  
    codice_fiscale char(40) not null,  
    grado char(40) not null,  
    constraint ID_CUOCO_ID primary key (codice_fiscale_cuoco),  
    constraint FKregistra_cuoco_ID unique (codice_fiscale));
```

```
create table DISPENSA (  
    idProdotto int not null,  
    prezzo float(1) not null,  
    data_scadenza date not null,  
    data_arrivo date not null,  
    temperatura_conservazione int not null,  
    nome char(40) not null,  
    idRistorante int not null,  
    quantita int not null,  
    constraint ID_DISPENSA_ID primary key (idProdotto));
```

```
create table intollerante (  
    nome char(40) not null,  
    nome_intolleranza char(40) not null,  
    constraint ID_intollerante_ID primary key (nome, nome_intolleranza));
```

```
create table INTOLLERANZA (  
    nome_intolleranza char(40) not null,  
    constraint ID_INTOLLERANZA_ID primary key (nome_intolleranza));
```



```
create table MENU (  
    prezzo int not null,  
    nome_portata char(40) not null,  
    id_portata int not null,  
    descrizione char(50) not null,  
    tempo_preparazione int not null,  
    constraint ID_MENU_ID primary key (id_portata));
```

```
create table ORDINE (  
    idOrdine int not null,  
    data date not null,  
    orario char(40) not null,  
    codice_fiscale_cameriere char(16) not null,  
    idRistorante int not null,  
    numero_tavolo int not null,  
    constraint ID_ORDINE_ID primary key (idOrdine));
```

```
create table prepara (  
    codice_fiscale_cuoco char(16) not null,  
    idPreparazione int not null,  
    constraint ID_prepara_ID primary key (idPreparazione, codice_fiscale_cuoco));
```

```
create table PREPARAZIONE (  
    idPreparazione int not null,  
    pasto char(40) not null,  
    descrizione char(50) not null,  
    data date not null,  
    id_portata int not null,  
    constraint ID_PREPARAZIONE_ID primary key (idPreparazione));
```



```
create table ricettario (  
    idProdotto int not null,  
    id_portata int not null,  
    constraint ID_ricettario_ID primary key (id_portata, idProdotto));
```

```
create table RISTORANTE (  
    idRistorante int not null,  
    nomeRistorante char(40) not null,  
    via char(40) not null,  
    numero_civico char(3) not null,  
    citta char(40) not null,  
    constraint ID_RISTORANTE_ID primary key (idRistorante));
```

```
create table STORICO_DIPENDENTI (  
    codice_fiscale char(40) not null,  
    nome char(40) not null,  
    cognome char(40) not null,  
    data_stipula_contratto date not null,  
    data_fine_contratto date,  
    constraint ID_STORICO_DIPENDENTI_ID primary key (codice_fiscale));
```

```
create table TAVOLO (  
    idRistorante int not null,  
    prenotato char not null,  
    max_persone int not null,  
    numero_tavolo int not null,  
    constraint ID_TAVOLO_ID primary key (idRistorante, numero_tavolo));
```

3.9 Traduzione delle operazioni in query SQL

01: inserimento ordine

```
INSERT INTO `ORDINE` VALUES ('0', '12/02/2024', '12:05', 'codice_fiscale', '0', '22');
```

02: visualizzare uno scontrino in una data specifica

```
SELECT totale FROM `CONTO` WHERE data='12/03/2024';
```

03: generare uno scontrino

```
INSERT INTO `CONTO` VALUES ('40', '0', '0', '12/02/2024');
```

04: prenotare un tavolo

```
UPDATE `TAVOLO` SET `prenotato` = '1' WHERE (`idRistorante` = '0') and (`numero_tavolo` = '22');
```

05: assegnare preparazioni ai cuochi

```
INSERT INTO `PREPARAZIONE` VALUES ('0', 'pranzo', 'descrizione', '12/02/2024', '0');
```

```
INSERT INTO `prepara` VALUES ('codice_fiscale', '0');
```

06: inserimento di nuovi prodotti nella dispensa

```
INSERT INTO DISPENSA values ('0', '10', '12/04/2024', '10/04/2024', '25', 'Broccolo', '0');
```

07: modificare le quantità rimanenti di un alimento nella dispensa

```
UPDATE DISPENSA` SET `quantita` = '20' WHERE (`idProdotto` = '0');
```

08: visualizzare le mansioni di un cuoco nella giornata

```
SELECT * FROM PREPARAZIONE WHERE data='12/02/24'
```

09: rinnovare il contratto ad un dipendente

```
UPDATE `STORICO_DIPENDENTI` SET `data_fine_contratto` = '12/02/2026' WHERE  
(`codice_fiscale` = 'codice_fiscale');
```

010: modificare uno o più piatti nel menù

```
UPDATE `MENU` SET `prezzo` = '25' WHERE (`id_portata` = '0');
```

011: aumentare di grado un cuoco

```
UPDATE `CUOCO` SET `grado` = 'chef' WHERE (`codice_fiscale_cuoco` = 'codice_fiscale');
```

012: assumere un nuovo dipendente

```
INSERT INTO `STORICO_DIPENDENTI` VALUES ('DIH294IDFHGO', 'marco', 'rossi',  
'12/02/2024', '12/02/2025');
```

```
INSERT INTO `CAMERIERE` VALUES ('DIH294IDFHGO', 'DIH294IDFHGO');
```

013: modificare il prezzo di vendita di un alimento nella dispensa

```
UPDATE `DISPENSA` SET `prezzo` = '25' WHERE (`idProdotto` = '0');
```

014: ricercare una preparazione fatta in una data specifica

```
SELECT * FROM PREPARAZIONE WHERE data='12/02/2024'
```

015: visualizzare il fatturato totale annuo

```
SELECT SUM(totale) AS totale_fatturato_annuo FROM CONTO WHERE data='12/02/2024';
```

016: aggiungere coperti ad un specifico tavolo

```
UPDATE `TAVOLO` SET `max_persone` = '5' WHERE (`idRistorante` = '0') and (`numero_tavolo`  
= '22');
```

017: ricercare un vecchio dipendente nello storico dipendenti

```
SELECT nome, cognome FROM STORICO_DIPENDENTI WHERE nome='Marco' AND  
cognome='ROSSI'
```

018: Visualizzare i top 3 piatti ordinati dal menù

```
SELECT CM.id_portata, M.nome_portata, COUNT(CM.idOrdine) as ordini  
FROM composto_menu CM JOIN MENU M ON CM.id_portata = M.id_portata  
GROUP BY CM.id_portata, M.nome_portata  
ORDER BY ordini DESC  
LIMIT 3;
```

019: Visualizzare la media dell'incasso mensile

```
SELECT AVG(totale) AS media  
FROM CONTO  
WHERE data BETWEEN '2024-01-01' AND '2024-01-30';
```

020: Visualizzare i primi 4 cuochi con meno preparazioni assegnate

```
SELECT P.codice_fiscale_cuoco, COUNT(P.idPreparazione) as preparazioni  
FROM prepara P JOIN CUOCO C ON C.codice_fiscale_cuoco = P.codice_fiscale_cuoco  
GROUP BY P.codice_fiscale_cuoco  
ORDER BY preparazioni DESC  
LIMIT 4
```