Elaborato per il corso di Basi di dati

A.A 2024/2025

**Progetto di una base di dati per la gestione di un ristorante di sushi**



Alice Beccati - 0001128202 (alice.beccati@studio.unibo.it)

Matteo Monari – 0001078461 (matteo.monari6@studio.unibo.it)

**Introduzione** 3

**Analisi dei requisiti**

* viste: clienti e proprietario 4
* Requisiti in linguaggio naturale 4
* estrazione dei concetti principali 5

**Progettazione Concettuale**

* Schema scheletro 6
* Sviluppo e affinamenti proposti 7
* Schema concettuale finale 10

**Progettazione logica**

* Stima del volume dei dati 11
* Descrizione delle operazioni principali e stima della loro frequenza 12
* Schemi di navigazione e tabelle degli accessi 13
* Ristrutturazione dello schema 20
* Analisi delle ridondanze 22
* Traduzione di entità e associazioni in relazioni 25
* Schema relazionale finale 27
* Traduzione delle operazioni in query SQL 28

**Progettazione dell'applicazione**

* Descrizione dell'architettura dell'applicazione 31

**Introduzione**

L’obiettivo di questo progetto è la realizzazione di un sistema informativo per supportare in maniera efficace la gestione operativa di un ristorante giapponese di sushi, aperto a inizio anno, con particolare attenzione alle dinamiche legate alla modalità di ordinazione All You Can Eat (AYCE).

Il sistema proposto consente di digitalizzare e strutturare i principali processi del ristorante: prenotazioni dei tavoli, gestione degli ordini, monitoraggio delle attività del personale e registrazione dei pagamenti. Un aspetto fondamentale riguarda la gestione del menù, pensata per essere compatibile sia con la modalità classica “alla carta”, sia con quella AYCE, che presenta vincoli quantitativi specifici per alcuni piatti selezionati (massimo 2 porzioni per persona).

Successivamente alla scelta di tipologia di menù preferita, valida per tutto il tavolo.

I clienti accedono al menù tramite QR code, da cui possono visualizzare in tempo reale i piatti disponibili, ciascuno dei quali è corredato da nome, descrizione, immagine e indicazioni su prezzo, ingredienti e di conseguenza allergeni, che in base all’alimento in cui si trovano possono essere più o meno concentrati e quindi risultare più o meno letali per un cliente allergico.

Le pietanze che i clienti intendono ordinare verranno, mano a mano, inserite nei vari ordini.

I quali saranno tenuti in memoria, in modo da poter controllare quali piatti si erano scelti e quali devono ancora arrivare, ma solo fino al momento in cui il tavolo risulta occupato; dunque, non si è ancora effettuato il pagamento. Successivamente il carrello con i vari ordini viene azzerato.

I ruoli all’interno dello staff del ristorante si dividono in chef e camerieri.

I primi si occupano della preparazione dei piatti, eventualmente anche collaborando tra loro. I camerieri, invece, consegnano i piatti che appartengono agli ordini ai tavoli corrispondenti.

Ogni dipendente effettua dei turni di lavoro in una certa data e in un certo turno, che può essere pranzo, cena o entrambi.

All’atto del pagamento viene generato uno scontrino con importo unico per tutto il tavolo, sul quale può essere applicato uno sconto in base al numero di persone che lo occupavano (es numero persone > 10 → 10% di sconto).

Successivamente al pagamento le informazioni su quel tavolo non interessano più, ma viene mantenuto uno storico soltanto dei pagamenti, in modo da permettere future statistiche sull’andamento del ristorante.

Sono previste funzionalità specifiche per l’amministratore, che ha la possibilità di monitorare tali statistiche (es. tavoli più prenotati, guadagni giornalieri, personale più attivo), gestire le scontistiche, mantenere aggiornato il menù e registrare le prenotazioni prese per via telefonica dai clienti.

È possibile prenotare anche più tavoli con una sola ordinazione. Le informazioni richieste durante l’operazione sono: la data, il turno, l’orario ed un nominativo.

**Analisi dei requisiti**

**Viste**

Le due tipologie di utenti che usufruiscono del sistema sono: il proprietario e i clienti.

Proprietario → Accede per aggiungere, rimuovere o modificare i piatti che appartengono al menu e che vengono visualizzati durante la fase di ordinazione.

Allo stesso modo può accedere alle informazioni che riguardano i dipendenti.

Può registrare o disdire una prenotazione.

Ha inoltre accesso ad informazioni di tipo statistico per visualizzare l’andamento del ristorante e la performance del personale.

Cliente → Un cliente si interfaccia al sistema tramite un Qrcode, che permette di visualizzare il menù, effettuare gli ordini e consultare il carrello contenente i piatti ordinati.

(Le prenotazioni vengono fatte esclusivamente per via telefonica)

**Requisiti in linguaggio naturale**

Si vuole realizzare un sistema per la gestione dell’attività di un ristorante di sushi.

Gli aspetti da esaminare sono:

* **ordinazioni dei clienti**

I clienti visualizzano il menu tramite Qrcode. La scelta dei menù si divide in : menù alla carta e AYCE (che ha un prezzo fisso e di cui si pagherà solo dolci e bevande), la tipologia deve essere uguale per tutti i clienti del tavolo, i quali vengono distinti con uno username.

Se il menu scelto è l’opzione AYCE, alcuni piatti selezionati possono essere ordinati massimo 2 porzioni a persona, quindi per un totale pari al doppio delle persone sedute al tavolo.

I piatti che possono essere ordinati presentano: un nome, una descrizione, gli ingredienti, una foto e un prezzo.

Ogni piatto è formato da un insieme di ingredienti e di conseguenza può avere degli allergeni.

Si vuole mantenere lo storico degli ordini precedenti del tavolo in un carrello, che verrà eliminato nel momento in cui la tavolata va a pagare il conto.

* **organizzazione personale interno**

Il personale si divide in chef e camerieri. Gli chef si occupano della preparazione dei piatti, mentre i camerieri consegnano gli ordini nei tavoli.

Le informazioni principali per i dipendenti sono: nome, ID e il turno a cui sono associati.

* **prenotazioni**

Lo stato del tavolo può essere “libero” o “prenotato/occupato”

Un cliente si occupa di effettuare una prenotazione, specificando data, orario, turno e indicando la quantità di tavoli che desidera riservare.

* **pagamento**

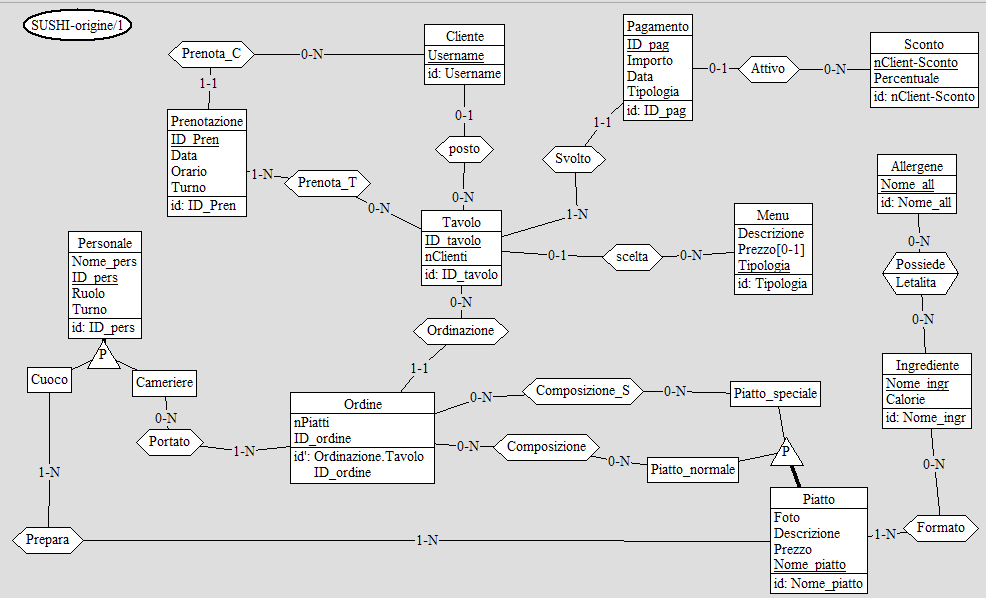
Il pagamento viene effettuato dal tavolo, con la gestione di un eventuale sconto (si applica in base al numero di persone del tavolo).

Viene memorizzato il guadagno delle varie giornate.

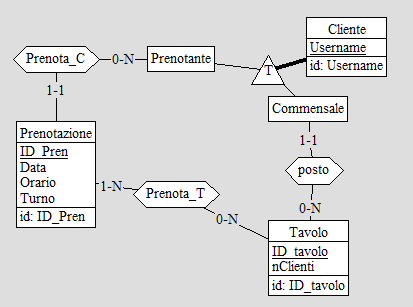
**estrazione dei concetti principali**

|  |  |  |
| --- | --- | --- |
| **Termine** | **Breve descrizione** | **Eventuale sinonimo** |
| Cliente | I clienti effettuano prenotazioni ed ordini, sono associati ad un tavolo | Commensale |
| Personale | Si occupa del servizio offerto, preparando e portando i piatti degli ordini | Dipendenti, staff |
| Ordine | È composto da diversi piatti ordinati dai clienti di un tavolo |  |
| Piatto | Preparato da uno chef con diversi ingredienti che possono contenere degli allergeni | Pietanza |
| Prenotazione | Effettuata da un cliente, in una data specifica. Può riguardare uno o più tavoli |  |
| Pagamento | Un tavolo viene liberato successivamente al pagamento del conto, sul quale si può applicare uno sconto | Saldo |

**Progettazione concettuale**

**Schema scheletro**

**Sviluppo e raffinamenti proposti**

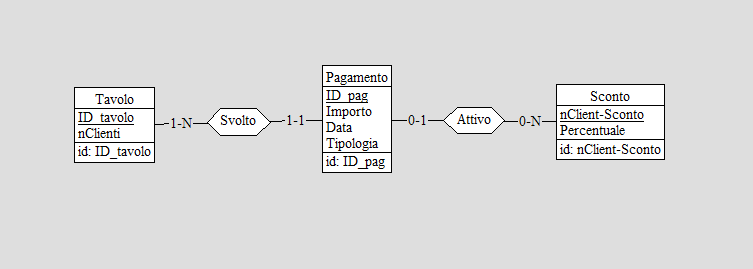
****

Con la seguente porzione di schema si intende modellare i clienti, che devono essere gestiti in base a che siano commensali, e quindi assegnarli ad un tavolo, o clienti prenotanti, che quindi lasciano il loro nominativo ai fini di una prenotazione.

La gerarchia risulta totale e sovrapposta, poiché un cliente prenotante può anch’esso occupare un posto ad un tavolo.

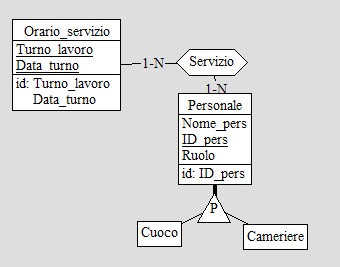
I prenotanti resteranno storicizzati del database per future rilevazioni a scopo statistico, mentre i soli commensali, una volta lasciato il locale, verranno eliminati.

Le prenotazioni sono storicizzate grazie ad un ID progressivo.

****

I pagamenti vengono considerati unici per ogni tavolo, ma storicizzati nel tempo, grazie all’ID\_pag che è un codice progressivo, per questo nello schema risulta che ad un tavolo possono far riferimento più pagamenti.

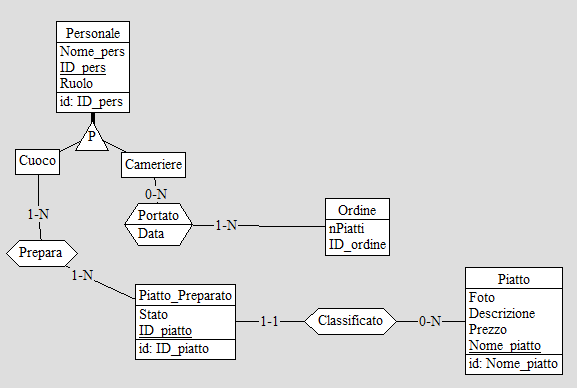
All’importo del pagamento può essere applicato uno sconto, la cui percentuale dipende dal numero di clienti del tavolo.

****

Nella porzione di schema sono modellati in maniera esaustiva, rispetto allo schema scheletro, i turni di servizio dei dipendenti. In questo modo è possibile mantenere uno storico dei turni di lavoro.

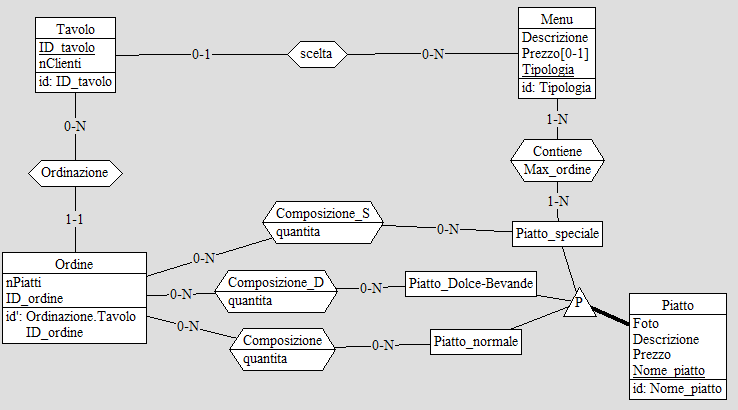
L’entità ORARIO\_SERVIZIO aggiunta ha una chiave primaria composta, in modo che sia possibile rappresentare il vincolo per cui un dipendente può effettuare più turni in una stessa data.

(Nello schema non è possibile rappresentarlo, i turni validi sono: pranzo e cena).

****

Nella gestione dei piatti preparati da un cuoco, lo shcema è stato raffinato per rappresentare efficientemente i singoli piatti, in preparazione e già preparati. PIATTO\_PREPARATO rappresenta i piatti fisici, mentre PIATTO è la classe generale.

Un ordine (che è composto da più piatti) può essere consegnato da più camerieri. Per mantere la data in cui un cameriere porta un determinato ordine viene salvata anche la data in cui questo avviene. Questo si rende necessario dal momento che gli ordini, una volta che il tavolo viene liberato, saranno eliminati.

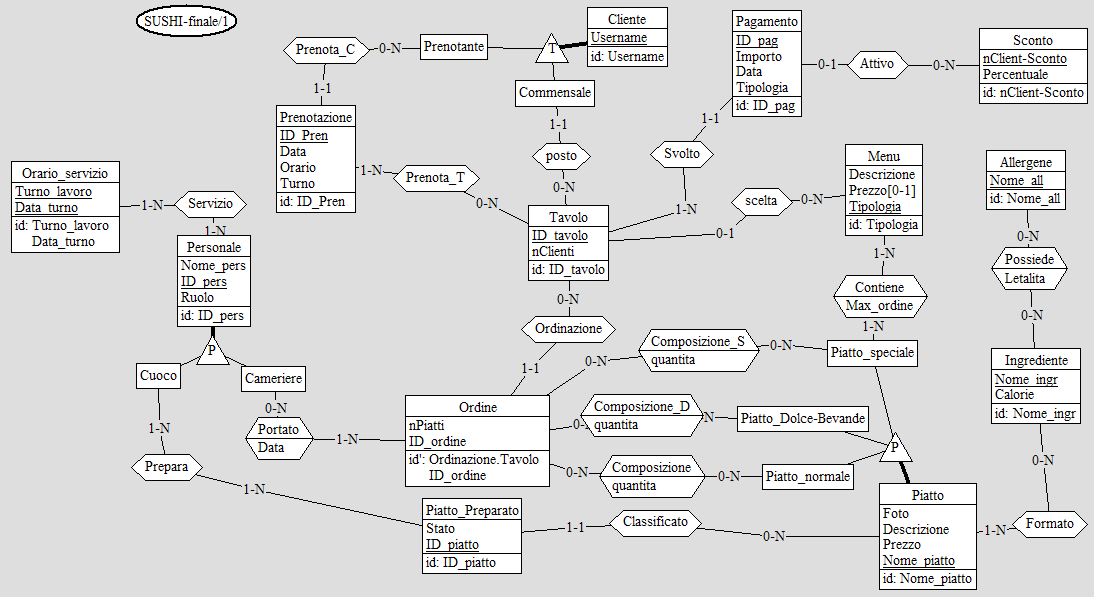
****

Per quanto riguarda la gestione dei piatti, oltre ai piatti speciali e normali è necessario distinguere i dolci e le bevande, che costituiscono una categoria a parte, poiché sarà necessario gestire il loro prezzo anche in caso di menù AYCE.

Per i piatti speciali, invece, viene introdotta un’associazione con menù in modo da permettere di modellare il vincolo per cui esso non può essere aggiunto ad un ordine oltre ad un numero massimo di volte.

Inoltre, nelle associazioni “composizione” sono state aggiunte le quantità per una simulazione più vicina alla realtà.

**Schema concettuale finale**



# Progettazione logica

**Stima del volume dei dati**

|  |  |  |
| --- | --- | --- |
| **Concetto** | **Costrutto** | **Volume** |
| Personale | E | 10 |
| Cameriere | E | 6 |
| Cuoco | E | 4 |
| Ordine | E | 160 |
| Ordinazione | A | 160 |
| Portato | A | 230 |
| Prepara | A | 4000 |
| Piatto\_Preparato | E | 3000 |
| Classificato | A | 3000 |
| Piatto | E | 150 |
| Formato | A | 750 |
| Ingrediente | E | 70 |
| Possiede | A | 140 |
| Allergene | E | 14 |
| Piatto\_speciale | E | 50 |
| Piatto\_normale | E | 100 |
| Composizione\_S | A | 750 |
| Composizione | A | 2250 |
| Contiene | A | 100 |
| menu | E | 2 |
| Scelta | A | 16 |
| Tavolo | E | 20 |
| Posto | A | 80 |
| Prenotante | E | 3.000 |
| Commensale | E | 12.000 |
| Cliente | E | 15.000 |
| Prenota\_C | A | 30.000 |
| Prenotazione | E | 30.000 |
| Prenota\_T | A | 35.000 |
| Svolto | A | 50.000 |
| Pagamento | E | 50.000 |
| Attivo | A | 20.000 |
| Sconto | E | 3 |
| Servizio | A | 7000 |
| Orario\_servizio | E | 720 |

**Descrizione delle operazioni principali e stima della loro frequenza**

1. Visualizzazione scelta del menu di un tavolo

(frequenza stimata: 60 volte al giorno)

1. Aggiunta di un piatto speciale all’ordine in caso di menù AYCE

(frequenza stimata: 3000 volte al giorno)

1. Visualizzare il totale di piatti ordinati da un tavolo

(frequenza stimata: 750 volte al giorno)

1. Registrare una prenotazione

(frequenza stimata: 20 volte al giorno)

1. Inserimento di un nuovo piatto

(frequenza stimata: 2 volte al mese)

1. Assunzione nuovo personale

(frequenza stimata: 4 volte all’anno)

1. Visualizzare piatti che prepara un cuoco

(frequenza stimata: 2 volte al giorno)

1. Spostamento di un cliente in un tavolo diverso presente nella stessa prenotazione (frequenza stimata: 1 volta alla settimana)
2. Applicare sconto ad un pagamento di un tavolo

(frequenza stimata: 6 volte al giorno)

1. Visualizzare cliente che ha effettuato il maggior numero di prenotazioni

(frequenza stimata: 1 volta al mese)

1. Visualizzare i 3 tavoli prenotati più frequentemente

(frequenza stimata: 1 volta al mese)

1. Visualizzare il mese con minor guadagno

(frequenza stimata: 1 volta all’anno)

1. Visualizzare cameriere che ha servito più tavoli in un giorno

(frequenza stimata: 1 volta al giorno)

**Schemi di navigazione e tabelle degli accessi**

**OP 1 Visualizzazione scelta del menu di un tavolo**

Per visualizzare il menù scelto da un tavolo si accede in lettura alle entità TAVOLO (per leggere il numero del tavolo) e MENU (per leggerne la tipologia) ed all’associazione che le collega.

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Tavolo | E | 1 | L |
| Scelta | A | 1 | L |
| menu | E | 1 | L |

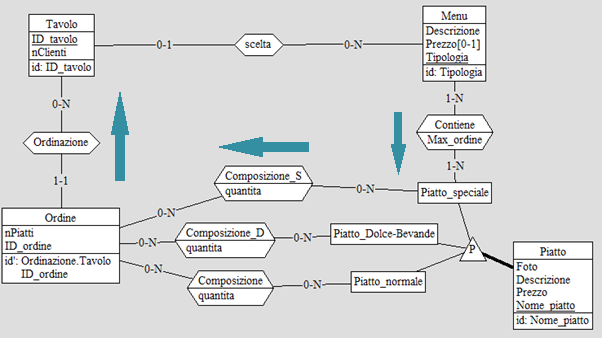
Totale 3 L → 180 accessi / gg

**OP 2 Aggiunta di un piatto speciale all’ordine in caso di menù AYCE**

Per aggiungere un piatto all’ordine occorre controllare prima che il numero di piatti speciali ordinati dal tavolo nei vari ordini non superi il limite consentito.

Se questo non succede si può procedere con l’aggiunta, supponendo che ordine e piatto speciale esistano già e che quindi non sia necessario scriverli.

Perciò ci sarà soltanto una scrittura su COMPOSIZIONE\_S ed una su ORDINE per aggiornare l’attributo ridondante “nPiatti”.

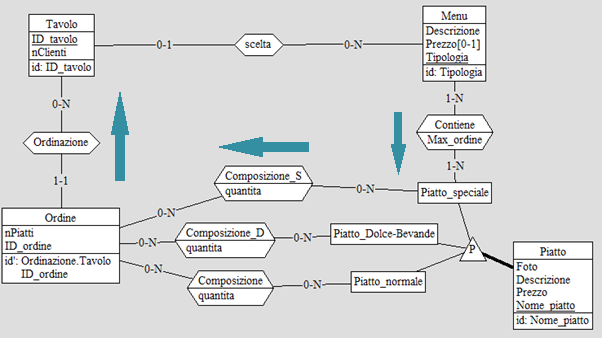


| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Contiene | A | 1 | L |
| Piatto speciale | E | 1 | L |
| Composizione\_S | A | 750/160=4.7 | L + 1S |
| Ordine | E | 8 | L + 1S |
| Ordinazione | A | 8 | L |
| Tavolo | E | 1 | L |

Totale 23.7L + 4S =27.7 → 83.100 accessi / gg

**OP 3 Visualizzare il totale di piatti ordinati da un tavolo**

Per visualizzare il totale di piatti ordinati da un singolo tavolo è sufficiente accedere in lettura al tavolo, all’ordine e alla loro associazione ordinazione, per un totale di 12.750 accessi al giorno



| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Tavolo | E | 1 | L |
| Ordinazione | A | 160/20=8 | L |
| Ordine | E | 8 | L |

Totale 17 L → 12.750 accessi / gg

**OP 4 Registrare una prenotazione**

Per registrare una nuova prenotazione si accede in lettura a Prenotazione per controllare che non esista già una prenotazione identica, se così non fosse, si accede in scrittura per aggiungere una nuova prenotazione e si aggiunge anche a Prenotante l’username di chi ha prenotato.

Successivamente si aggiornano anche le die associazione Prenota\_C e Prenota\_T

Immagine che contiene testo, schermata, diagramma, Piano

Il contenuto generato dall'IA potrebbe non essere corretto.che gestiscono rispettivamente la prenotazione del Prenotante e del tavolo

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Prenotazione | E | 30000 | L |
| Prenotazione | E | 1 | S |
| Prenota\_T | A | 35000 | L |
| Prenota\_T | A | 1.17 | S |
| Prenotante | E | 1 | S |
| Prenota\_C | A | 1 | S |

Totale 65000L+8.34S=65008.34  →1300166.8 accessi / gg

**OP 5 Inserimento di un nuovo piatto**

Per inserire un nuovo piatto vengono eseguiti due accessi in scrittura a Piatto e a Piatto normale o speciale in base alla tipologia del piatto che si vuole aggiungere

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Piatto | E | 1 | S |
| Piatto normale/speciale | E | 1 | S |

Totale  2S →8 accessi / mese

**OP 6 Assunzione nuovo personale**

Per assumere nuovo personale si procede con operazioni analoghe all’operazione 5, cioè si va ad aggiungere a Personale e Cuoco o Cameriere in riferimento al ruolo che esegue.

Totale 2S →16 accessi / anno

**OP 7 Visualizzare piatti che prepara un cuoco**

Si vuole visualizzare i piatti preparati da un cuoco quindi si accede in lettura a Cuoco, Piatto preparato e all’associazione Prepara.

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Cuoco | E | 1 | L |
| Prepara | A | 1000 | L |
| Piatto preparato | E | 1000 | L |

Totale  3L→6 accessi / gg

**OP 8 Spostamento di un cliente in un tavolo diverso presente nella stessa prenotazione**

Per spostare un cliente Commensale in un tavolo diverso presente nella stessa prenotazione è necessario leggere il posto in cui siede il cliente e l’associazione Prenota\_T che indica quali tavoli sono sotto la stessa prenotazione.

Trovato un secondo tavolo si aggiorna Posto con il nuovo posto del cliente e il tavolo che avrà una persona aggiunta (nClienti + 1).

Immagine che contiene testo, diagramma, schermata, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Cliente | E | 1 | L |
| Posto | A | 1 | L |
| Posto | A | 1 | S |
| Tavolo | E | 1 | L |
| Tavolo | E | 2 | S |
| Prenota\_T | A | 1 | L |

Totale 6S+4L→10 accessi / settimana

**OP 9 Applicare sconto ad un pagamento di un tavolo**

Applicare uno sconto ad un pagamento di un tavolo è un’operazione analoga all’operazione 2. In più se si vuole conoscere il tavolo di appartenenza del pagamento quindi si effettuano letture anche su Svolto e Tavolo.

Totale 1S+4L→36 accessi / gg

**OP 10 Visualizzare cliente che ha effettuato il maggior numero di prenotazioni**

Per visualizzare cliente Prenotante che ha effettuato il maggior numero di prenotazioni si legge il cliente da Prenotante si trova la prenotazione tramite una lettura dell’associazione e con l’ultima lettura si contano le prenotazioni in Prenotazione

Immagine che contiene testo, diagramma, schermata, Piano

Il contenuto generato dall'IA potrebbe non essere corretto.

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Prenotante | E | 15.000 | L |
| Prenota\_C | A | 30.000 | L |
| Prenotazione | E | 30.000 | L |

Totale 75.000L →75.000 accessi / mese

**OP 11 Visualizzare i 3 tavoli prenotati più frequentemente**

Per visualizzare i 3 tavoli prenotati più frequentemente si accede in lettura all’associazione Prenota\_T e si contano i tavoli in Tavolo.

|  |  |  |  |
| --- | --- | --- | --- |
| Prenota\_T | A | 35.000 | L |
| Tavolo | E | 20 | L |

Totale 35020L →  35020 accessi / mese

**OP 12 Visualizzare il mese con minor guadagno**

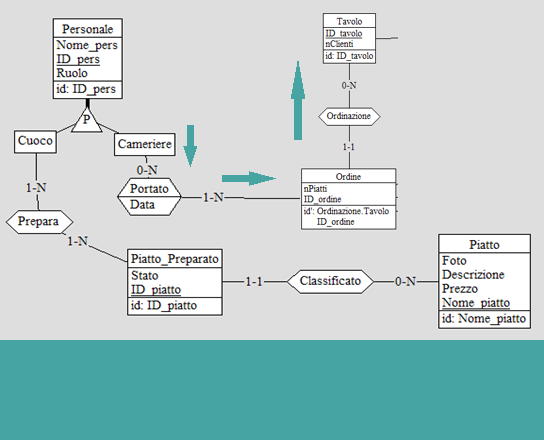
Per visualizzare il mese con minor guadagno accedo in lettura a Pagamento e seleziono il mese dall’attributo Data controllando che non ci sia da applicare uno sconto presente in Sconto. Se è presente uno sconto per quel pagamento allora verrà calcolato il nuovo importo

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Pagamento | E | 50.000 | L |
| Sconto | E | 20.000 | L |
| Attivo | A | 20.000 | L |

Totale 90.000L→ 90.000 accessi / anno

**OP 13 Visualizzare il cameriere che ha servito più tavoli in un giorno**

Per visualizzare il cameriere che ha servito più tavoli in un giorno si accede in lettura per visualizzare il cameriere poi si leggono gli ordini che ha portato in un determinato giorno da Ordine e l’associazione Portato (con attributo Data) e infine i tavoli che ha servito da Tavolo e dall’associazione Ordinazione



|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo |
| Cameriere | E | 6 | L |
| Portato | A | 230 | L |
| Ordine | E | 230 | L |
| Ordinazione | A | 230 | L |
| Tavolo | E | 230 | L |

Totale 926 →926 accessi / gg

**Ristrutturazione dello schema**

**Eliminazione delle gerarchie**

La gerarchia presente in Personale, che si divideva in Cuoco e Cameriere, è stata risolta applicando la tecnica del collasso verso l’alto, in quanto entrambi i sottotipi condividevano gli stessi attributi presenti nell’entità Personale. È stato quindi possibile eliminare le due entità figlie, introducendo un unico attributo Ruolo direttamente nell’entità Personale.

Anche la gerarchia presente nell’entità Piatto, che si divideva in Piatto Speciale, Piatto Normale e Piatto Dolce-Bevande, è stata risolta applicando la medesima tecnica del collasso verso l’alto. È stato infatti possibile aggiungere l’attributo Tipologia direttamente all’entità Piatto per distinguere i diversi sottotipi, eliminando così le entità figlie e unificando la gestione delle

associazioni (Composizione, Composizione\_S, Composizione\_D) in un’unica relazione.

Un analogo ragionamento è stato applicato anche all’entità Cliente, che inizialmente si divideva nei sottotipi Prenotante e Commensale. La gerarchia è stata quindi collassata verso l’alto, eliminando le entità figlie e gestendo la distinzione tra le due tipologie di clienti mediante l’aggiunta dell’attributo Attivo direttamente nell’entità Cliente. In questo modo, il cliente Prenotante viene storicizzato, mentre il cliente Commensale viene temporaneamente memorizzato e successivamente eliminato al termine del pasto.

**Scelta delle chiavi primarie**

Nello schema sono già evidenziate senza ambiguità tutte le chiavi primarie per la maggior parte delle entità, spesso vengono usati gli ID o nomi/username.

**Eliminazione degli identificatori esterni**

Nello schema E/R sono eliminate le seguenti relazioni:

Posto, importando un attributo opzionale a Cliente.

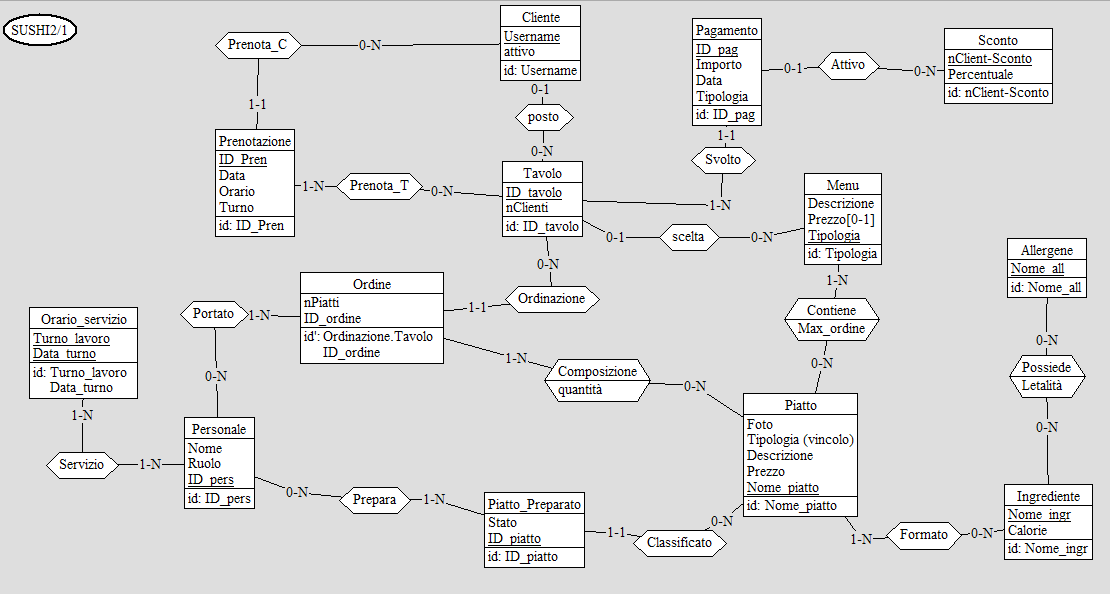
Prenota\_C, importando Username del cliente a prenotazione.

Ordinazione, importando ID\_tavolo a Ordine.

Classificato, importando in Piatto\_preparato il nome del piatto.

Scelta, importando in tavolo l’attributo opzionale Tipologia in quanto può anche essere libero.

Svolto e Attivo importando in agamento Id tavolo e l’attributo opzionale nClienti utili per avere uno sconto.



**Analisi delle ridondanze**

Si nota che l’attributo **nClienti** è un attributo ridondante in quanto è calcolabile visualizzando i clienti associati ad un determinato tavolo tramite l’associazione posto.

Inoltre, anche **nPiatti** risulta ridondante siccome è calcolabile dall’associazione Composizione tra Ordine e Piatto.

Questi attributi vanno ad influire sulle operazioni 2,3,8.  
  
**OP 2 Aggiunta di un piatto speciale all’ordine in caso di menù AYCE**

**CON RIDONDANZA nPiatti + nClienti**

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Contiene | A | 1 | L |
| Piatto speciale | E | 1 | L |
| Composizione\_S | A | 750/160=4.7 | L + 1S |
| Ordine | E | 8 | L + 1S |
| Ordinazione | A | 8 | L |
| Tavolo | E | 1 | L |

Totale 23.7L + 4S =27.7 → 83.100 accessi / gg

**CON RIDONDANZA nClienti (senza nPiatti)**

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Contiene | A | 1 | L |
| Piatto speciale | E | 1 | L |
| Composizione\_S | A | 750/160=4.7 | L + 1S |
| Ordine | E | 8 | L |
| Ordinazione | A | 8 | L |
| Tavolo | E | 1 | L |

Totale 23.7L + 1S =25.7 → 77.100 accessi / gg

**CON RIDONDANZA nPiatti (senza nClienti)**

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Cliente | E | 4 | L |
| Contiene | A | 1 | L |
| Piatto speciale | E | 1 | L |
| Composizione\_S | A | 750/160=4.7 | L + 1S |
| Ordine | E | 8 | L + 1S |
| Ordinazione | A | 8 | L |
| Tavolo | E | 1 | L |
| Posto | A | 80/20=4 | L |

Totale 31,7L + 2S =35,7 →107.100 accessi / gg

**SENZA RIDONDANZA (senza nClienti e nPiatti)**

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Clienti | E | 80/20=4 | L |
| Contiene | A | 1 | L |
| Piatto speciale | E | 1 | L |
| Composizione\_S | A | 750/160=4.7 | L + 1S |
| Ordine | E | 8 | L |
| Ordinazione | A | 8 | L |
| Tavolo | E | 1 | L |
| Posto | A | 80/20=4 | L |

Totale 31.7L + 1S =33.7 →101.100 accessi / gg

Si nota che la soluzione per aggiungere un piatto speciale al menù è la soluzione con ridondanza dell’attributo nClienti ma senza l’attributo nPiatti che risulta ottima (77.100 accessi al giorno) in quanto vengono fatte, una sola scrittura in composizione\_S e 23.7 letture.

**OP 3 Visualizzare il totale di piatti ordinati da un tavolo**

**CON RIDONDANZA di nPiatti**

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Tavolo | E | 1 | L |
| Ordinazione | A | 160/20=8 | L |
| Ordine | E | 8 | L |

Totale 17 L → 12.750 accessi / gg

**SENZA RIDONDANZA di nPiatti**

| Concetto | Costrutto | Accessi | Tipo |
| --- | --- | --- | --- |
| Tavolo | E | 1 | L |
| Ordinazione | A | 160/20=8 | L |
| Ordine | E | 8 | L |
| Composizione | A | 2250/160=14 | L |
| Composizione\_S | A | 4.7 | L |

Totale 35,7 L → 26,775 accessi / gg

Per visualizzare il totale di piatti ordinati da un tavolo la soluzione più veloce è il mantenimento dell’attributo ridondante nPiatti presente in Ordine in quanto si ha un totale di accessi di 12.750 accessi al giorno contro i 26.775 che si dovrebbero fare se si togliesse nPiatti da Ordine.

**OP 8 Spostamento di un cliente in un tavolo diverso presente nella stessa prenotazione**

**CON RIDONDANZA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L+1S |
| Posto | A | 1 | L |
| Posto | A | 1 | S |
| Tavolo | E | 1 | L |
| Tavolo | E | 2 | S |
| Prenota\_T | A | 1 | L |

Totale 4S+4L→12 accessi / settimana = 1.71 al giorno

**SENZA RIDONDANZA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Concetto** | **Costrutto** | **Accessi** | **Tipo** |
| Cliente | E | 1 | L+1S |
| Posto | A | 1 | L |
| Posto | A | 1 | S |
| Tavolo | E | 1 | L |
| Prenota\_T | A | 1 | L |

Totale 2S+4L→8 accessi / settimana = 1.14 al giorno

Per spostare un cliente in un tavolo diverso presente nella stessa prenotazione la soluzione ottimale si ha rimuovendo l’attributo ridondante nClienti dall’entità Tavolo in quanto si procede a modificare con un accesso in scrittura solo il valore in Posto e non anche l’attributo in Tavolo, di conseguenza si ha un totale di accessi nella versione senza ridondanza pari a 8 accessi a settimana contro i 12 che si avrebbero con la versione con ridondanza.

**Conclusioni**

Dopo l’analisi delle ridondanze, facendo le somme del numero di accessi nel caso in cui: vengano mantenuti entrambi gli attributi ridondanti, venga mantenuto solo nClienti, venga mantenuto solo nPiatti o vengano eliminate entrambe.

Risulta utile mantenere l’attributo ridondante nClienti da Tavolo e mantenere l’attributo nPiatti in Ordine.

**Traduzione di entità e associazioni in relazioni**

**Clienti(Username, attivo, ID\_tavolo\*)  
FK: Cliente(ID\_tavolo) REF Tavolo(ID\_tavolo)**

**Tavoli**(*ID\_tavolo*, nClienti, *Tipologia\**)  
FK: Tavolo(Tipologia) \* REF menu(Tipologia)

**Prenotazioni**(*ID\_Pren*, Data, Orario, Turno, Username)  
FK: Prenotazione(Username) REF Cliente(Username)

**Prenota\_T**(*ID\_Pren*, ID\_tavolo)  
FK: Prenota\_T(ID\_Pren) REF Prenotazione(ID\_Pren)  
FK: Prenota\_T(ID\_tavolo) REF Tavolo(ID\_tavolo)

**Pagamenti**(*ID\_pag*, Importo, Data, Tipologia, ID\_tavolo, nClient\_Sconto\*)  
FK: Pagamento(ID\_tavolo) REF Tavolo(ID\_tavolo)  
FK: Pagamento(cClient\_Sconto) \* REF Sconto(cClient\_Sconto)

**Sconti**(*nClient\_Sconto*, Percentuale)

**Personale**(*ID\_personale*, Nome\_pers, Ruolo)

**Ordini**((*ID\_ordine*, ID\_tavolo), *nPiatti*)  
FK: Ordine(ID\_tavolo) REF Tavolo(ID\_tavolo)

**Portato**((*ID\_ordine*, ID\_tavolo), ID\_personale, data)  
FK: Trasporto(ID\_ordine, ID\_tavolo) REF Ordine(ID\_ordine,ID\_tavolo)  
FK: Trasporto(ID\_personale) REF Personale(ID\_personale)

**Prepara**(*ID\_personale*, *ID\_piatto*)  
FK: Prepara(ID\_personale) REF Personale(ID\_personale)  
FK: Prepara(ID\_piatto) REF Piatto(ID\_piatto)

**Piatti\_Preparati**(*ID\_piatto*, Nome\_piatto, Stato)  
FK: Piatto\_Preparato(Nome\_piatto) REF Piatto(Nome\_piatto)

**Piatti**(*Nome\_piatto*, *Foto*, Tipologia, *Descrizione*, *Prezzo*)

**Composizioni**((*ID\_tavolo*, *ID\_ordine)*, *Nome\_piatto* *,Quantità*)  
FK: Composizione(ID\_tavolo, ID\_ordine) REF Ordine(ID\_tavolo,ID\_ordine)  
FK: Composizione(Nome\_piatto) REF Piatto(Nome\_piatto)

**menu**(*Tipologia*, *Descrizione*, *Prezzo*\*)

**Contiene**(*Nome\_piatto*, *Tipologia*, *Max\_ordine*)  
FK: Contiene(Nome\_piatto) REF Piatto(Nome\_piatto)  
FK: Contiene(Tipologia) REF menu(Tipologia)

**Formato**(*Nome\_ingr*, *Nome\_piatto*)  
FK: Formato(Nome\_ingr) REF Ingredienti(Nome\_ingr)  
FK: Formato(Nome\_piatto) REF Piatto(Nome\_piatto)

**Ingredienti**(*Nome\_ingr*, Calorie)

**Allergeni**(*Nome\_all*)

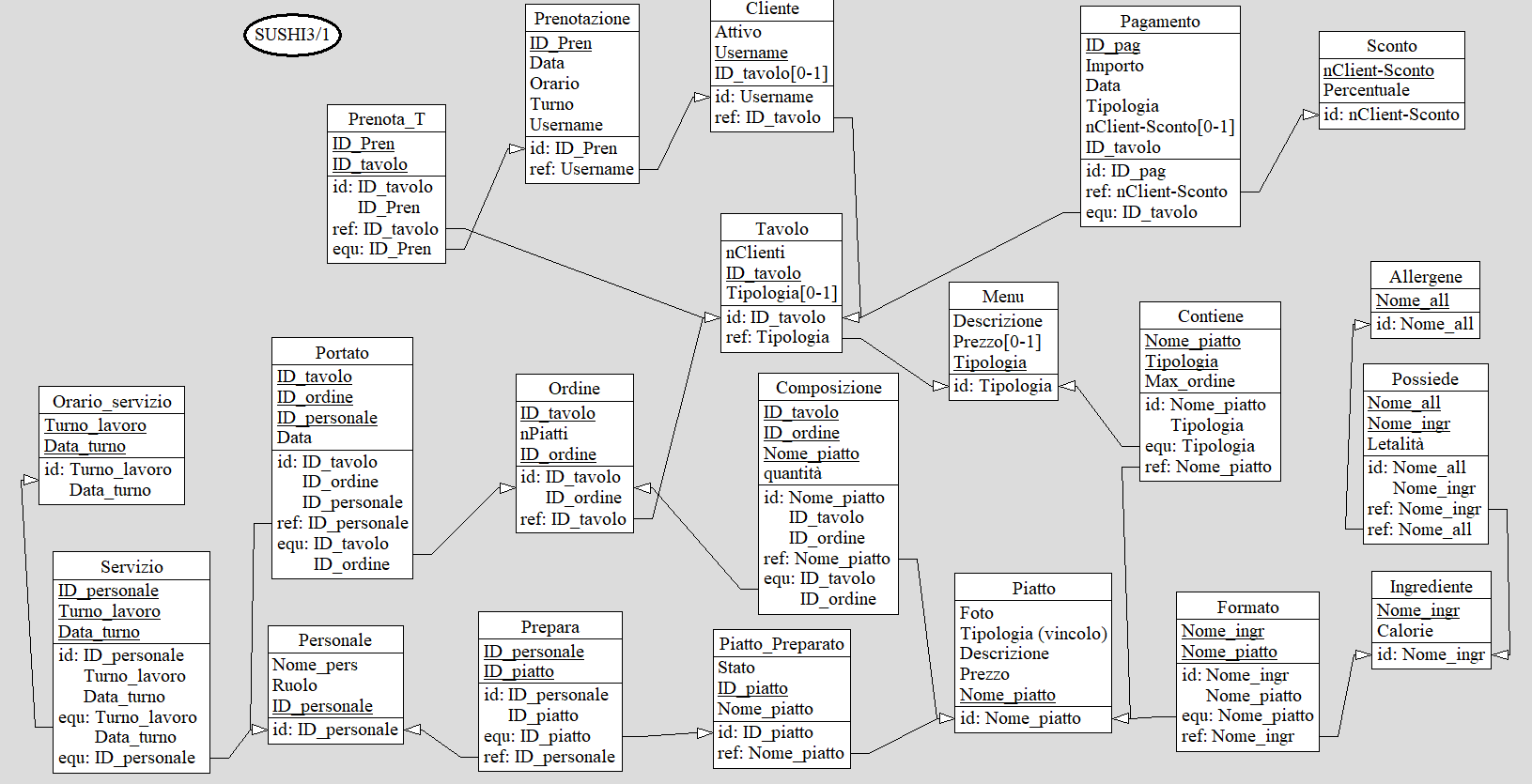
**Possiede**(*Nome\_ingr*, *Nome\_all*)  
FK: Possiede(Nome\_ingr) REF Ingredienti(Nome\_ingr)  
FK: Possiede(Nome\_all) REF Allergene(Nome\_all)

**Servizio**(ID\_personale,(Turno\_lavoro,Data\_turno))

FK: Servizio(Turno\_lavoro,Data\_turno) REF Orario\_servizio(Turno\_lavoro,Data\_turno)  
FK: Servizio(ID\_personale) REF Personale(ID\_personale)

**Orario\_servizio**(Turno\_lavoro,Data\_turno)

**Schema relazionale finale**



**Traduzione delle operazioni in query SQL**

**1 Visualizzazione scelta del menu di un tavolo**

SELECT t.ID\_tavolo, t.Tipologia, m.Descrizione

FROM Tavolo t

JOIN menu m ON t.Tipologia = m.Tipologia

WHERE t.ID\_tavolo = 16;

**2 Aggiunta di un piatto speciale/normale all’ordine**

INSERT INTO Composizione (ID\_tavolo, ID\_ordine, Nome\_piatto, Quantita)

SELECT 17, 43, 'TartareTonno', 1

WHERE (

    (SELECT p.Tipologia

     FROM Piatto p

     WHERE p.Nome\_piatto = 'TartareTonno') != 'Special'

    OR

    (

      (SELECT p.Tipologia

       FROM Piatto p

       WHERE p.Nome\_piatto = 'TartareTonno') = 'Special'

      AND

      (SELECT COALESCE(SUM(c.Quantita), 0)

       FROM Composizione c

       WHERE c.ID\_tavolo = 16

         AND c.Nome\_piatto = 'TartareTonno') <

      (SELECT Max\_ordine

       FROM Contiene co

       WHERE co.Nome\_piatto = 'TartareTonno'

       LIMIT 1)

    )

);

**3 Visualizzare il totale di piatti speciali ordinati da un tavolo**

SELECT SUM(c.Quantita), c.ID\_tavolo

FROM Composizione c

JOIN Piatto p ON p.Nome\_piatto = c.Nome\_piatto

WHERE c.ID\_tavolo = '17'

 AND p.Tipologia = 'Special';

**4 Registrare una prenotazione**

INSERT INTO Prenotazione (ID\_Pren, Data, Orario, Turno, Username)

VALUES(30,'2025-09-18', '13:30:00', 'Pranzo', 'Giovanni')

**5 Inserimento di un nuovo piatto** procedimento analogo a 4

**6 Assunzione nuovo personale** procedimento analogo a 4

**7 Visualizzare piatti che prepara uno cuoco**

SELECT p.Nome\_Piatto

FROM Prepara pr

JOIN Piatto\_Preparato pp ON pp.ID\_Piatto = pr.ID\_Piatto

JOIN Piatto p ON pp.Nome\_piatto = p.Nome\_piatto

JOIN Personale pe ON pr.ID\_Personale = pe.ID\_Personale

WHERE pr.ID\_Personale = 7 AND pe.Ruolo = 'Cuoco' AND pp.Stato='Consegnato'

**8 Spostamento di un cliente in un tavolo diverso presente nella stessa prenotazione**

UPDATE Cliente

SET ID\_tavolo = '1'

WHERE Username = 'user1'

AND ID\_tavolo = '5'

AND EXISTS (

SELECT 1

FROM Prenota\_T pt1

JOIN Prenota\_T pt2 ON pt1.ID\_Pren = pt2.ID\_Pren

WHERE pt1.ID\_tavolo = '1'

AND pt2.ID\_tavolo = '5'

);

**9 Applicare sconto ad un pagamento di un tavolo**

UPDATE Pagamento p

JOIN Sconto s ON p.nClient\_Sconto = s.nClient\_Sconto

SET p.Importo = ROUND(p.Importo \* (1 - s.Percentuale / 100), 2)

WHERE p.ID\_pag = 1;

Si precisa che, ove opportuno, sarebbe stato possibile fare uso del comando WITH TIES per gestire eventuali situazioni di pareggio. Tuttavia, il database in uso non implementa tale funzionalità, pertanto non è stato possibile adottarla nelle query proposte.

**10 Visualizzare cliente che ha effettuato il maggior numero di prenotazioni**

SELECT p.Username, COUNT(\*) AS nPren

FROM Prenotazione p

GROUP BY p.Username

HAVING COUNT(\*) = (

    SELECT MAX(conteggio)

    FROM (

        SELECT COUNT(\*) AS conteggio

        FROM Prenotazione

        GROUP BY Username

    ) AS sott

);

**11 Visualizzare i 3 tavoli prenotati più frequentemente**

SELECT sub.ID\_tavolo, sub.nPren

FROM (

    SELECT pt.ID\_tavolo, COUNT(\*) AS nPren,

           DENSE\_RANK() OVER (ORDER BY COUNT(\*) DESC) AS posizione

    FROM Prenota\_T pt

    GROUP BY pt.ID\_tavolo

) AS sub

WHERE sub.posizione <= 3;

**12 Visualizzare il mese con minor guadagno**

SELECT

DATE\_FORMAT(p.Data, '%Y-%m') AS mese,

SUM(

p.Importo

\* (1 - COALESCE(sc.Percentuale, 0) / 100)

) AS guadagno\_netto

FROM Pagamento p

LEFT JOIN Sconto sc

ON p.nClient\_Sconto = sc.nClient\_Sconto

GROUP BY mese

ORDER BY guadagno\_netto ASC

LIMIT 1;

**13 Visualizzare il cameriere che ha servito più tavoli in un giorno**

SELECT

p.Nome\_pers,

po.Data,

COUNT(DISTINCT po.ID\_tavolo) AS tavoli\_serviti

FROM Portato po

JOIN Personale p ON po.ID\_personale = p.ID\_personale

GROUP BY p.ID\_personale, po.Data

HAVING COUNT(DISTINCT po.ID\_tavolo) = (

SELECT

MAX(tavoli\_per\_giorno)

FROM (

SELECT

COUNT(DISTINCT po2.ID\_tavolo) AS tavoli\_per\_giorno

FROM Portato po2

GROUP BY po2.ID\_personale, po2.Data

) AS subquery

)

ORDER BY po.Data, p.Nome\_pers;

**Descrizione dell'architettura dell'applicazione**

L'applicazione web si basa su TiDB, un sistema di gestione di database relazionali distribuito open source che offre compatibilità con MySQL. Questa scelta ci ha permesso di lavorare in modalità remota sul server.

L'accesso e la gestione del database avvengono tramite MySQL Workbench, sfruttando la compatibilità di TiDB con gli strumenti MySQL.

Per la realizzazione dell’applicazione web ci siamo serviti di **Django** (framework web open source scritto in Python, progettato per facilitare lo sviluppo rapido di applicazioni web sicure e manutenibili).

Django segue la logica **MTV (Model-Template-View)**, che separa la logica dei dati (Model), la presentazione (Template) e la logica di controllo (View).

Immagine che contiene testo, verdura, Gruppo di alimenti, Fast food

Il contenuto generato dall'IA potrebbe non essere corretto.**Immagine che contiene testo, schermata, cibo

Il contenuto generato dall'IA potrebbe non essere corretto.**Dopo aver creato la connessione con il database ed aver di ottenuto così la parte di Model, ci siamo dedicati alla realizzazione delle pagine html (Template) e della scrittura del codice python per la gestione delle operazioni sul database (View)

Immagine che contiene testo, schermata, software, Pagina Web

Il contenuto generato dall'IA potrebbe non essere corretto.

Nella parte di sito dedicata agli utenti vengono visualizzati tutti i piatti disponibili nel menù, tramite lettura apposita della tabella PIATTI nel database.

Sarà possibile aggiungere i piatti che si desira ordinare in un carrello.

**Immagine che contiene testo, schermata, numero, Carattere

Il contenuto generato dall'IA potrebbe non essere corretto.**Una volta confermato ed inviato l’ordine esso verrà registrato e sarà possibile visualizzare tutti i piatti contenuti negli ordini effettuati dal tavolo.

**Immagine che contiene testo, design, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.**

Nella parte dedicata alla vista dell’amministratore sarà possibile eseguire alcune operazioni di gestione del ristorante, che verranno svolte dal proprietario. Per esempio :

* L’inserimento di una nuova prenotazione, che permette di aggiungere al database anche il cliente che sta prenotando, nel caso in cui esso non sia già stato precedentemente registrato.
* Inserire un piatto, con relative informazioni, previo controllo che esso non sia già presente. Viene gestito anche l’inserimento di un’immagine.
* Inserimento di un nuovo dipendente, se non già presente.
* Possibilità di visualizzare tutti i dipendenti assunti e scegliere di eliminarne uno dal sistema.