



UNIVERSITATEA TEHNICĂ "GH ASACHI" IAȘI FACULTATEA AUTOMATICĂ ȘI
CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
DISCIPLINA BAZE DE DATE PROIECT

Sistem de Gestionare a Rezervărilor pentru Restaurante

**Coordonator,
Cătălin Mironeanu**

**Student,
Fundueanu-Constantin Stefan
Grupa 1405A**

Iași, 2024



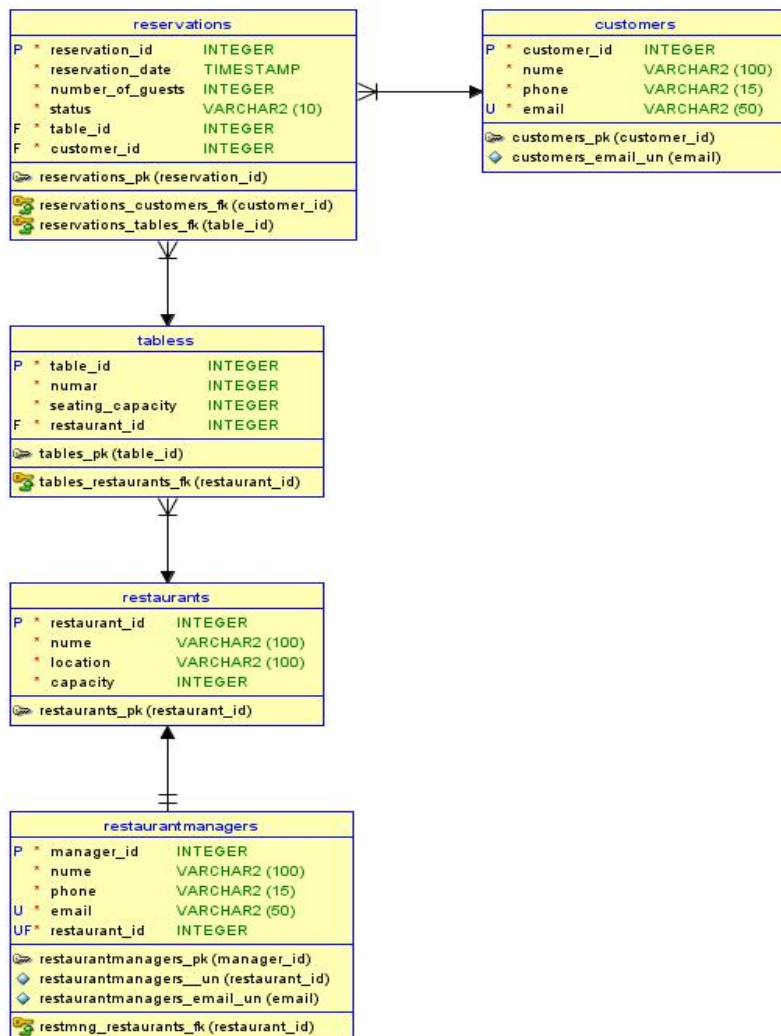
Introducere

Descrierea Problemei

Industria restaurantelor se confruntă adesea cu provocări legate de gestionarea eficientă a rezervărilor și a capacității de locuri. Problemele comune includ suprasolicitarea resurselor, erori în alocarea meselor și dificultăți în urmărirea managerilor și a personalului asociat fiecărui restaurant. O soluție informatică bine concepută poate îmbunătăți semnificativ aceste procese, reducând erorile și sporind satisfacția clienților.

Structura și inter-relationarea tabelelor

Diagrama ER



Scopul proiectului

Scopul acestui proiect este de a dezvolta o bază de date relațională pentru gestionarea rezervărilor într-un lanț de restaurante. Sistemul va permite:

- ◆ Gestionarea eficientă a rezervărilor de către clienți.
- ◆ Asigurarea conformității rezervărilor cu capacitatea maximă a meselor și restaurantelor.
- ◆ Administrarea managerilor de restaurante, asigurându-se că fiecare restaurant are un singur manager.
- ◆ Îmbunătățirea urmăririi și raportarea activităților de rezervare și utilizare a



resurselor.

Structura Bazei de Date

Baza de date este structurată pentru a satisface nevoile de gestionare a rezervărilor și a capacității restaurantelor. Structura bazei de date include următoarele tabele principale:

Tabela customers

- customer_id: Identificator unic pentru fiecare client.
- nume: Numele clientului.
- phone: Numărul de telefon al clientului.
- email: Adresa de email a clientului.

Tabela reservations

- reservation_id: Identificator unic pentru fiecare rezervare.
- reservation_date: Data și ora rezervării.
- number_of_guests: Numărul de invitați pentru rezervare.
- status: Starea rezervării (de exemplu, Confirmed, Pending, Cancelled).
- table_id: Identificatorul mesei rezervate.
- customer_id: Identificatorul clientului care a făcut rezervarea.

Tabela restaurantmanagers

- manager_id: Identificator unic pentru fiecare manager.
- nume: Numele managerului.
- phone: Numărul de telefon al managerului.
- email: Adresa de email a managerului.
- restaurant_id: Identificatorul restaurantului pe care îl administrează managerul.

Tabela restaurants

- restaurant_id: Identificator unic pentru fiecare restaurant.
- nume: Numele restaurantului.



- location: Locația restaurantului.
- capacity: Capacitatea totală de locuri a restaurantului.

Tabela tabless

- table_id: Identificator unic pentru fiecare masă.
- numar: Numărul mesei.
- seating_capacity: Capacitatea de locuri a mesei.
- restaurant_id: Identificatorul restaurantului căruia îi aparține masa.

Relațiile dintre Tabele

Relațiile dintre tabele sunt definite prin constrângerile de cheie străină, care asigură integritatea referențială între tabele. În cadrul proiectului nostru, relațiile dintre tabele sunt următoarele:

customers și reservations:

Un client (customers) poate avea mai multe rezervări (reservations), dar fiecare rezervare este asociată unui singur client.

Relația este de tipul 1:n (unul la mulți).

restaurants și restaurantmanagers:

Un restaurant (restaurants) poate avea un singur manager (restaurantmanagers), iar un manager poate administra un singur

restaurant.

Relația este de tipul 1:1 (unu la unu).

restaurants și tabless:

Un restaurant (restaurants) poate avea mai multe mese (tabless), dar fiecare masă este asociată unui singur restaurant.



Relația este de tipul 1:n (unul la mulți).

tabless și reservations:

O masă (tabless) poate avea mai multe rezervări (reservations), dar fiecare rezervare este asociată unei singure mese.

Relația este de tipul 1:n (unul la mulți).

DESCRIEREA LOGICII STOCATE

În cadrul proiectului, pentru a evidenția aspectele practice ale acestuia, s-au utilizat următoarele:

❖ TRIGGERE:

- **trg_delete_customer:** Acest trigger asigură că, atunci când un client este șters, toate rezervările asociate acestuia sunt de asemenea șterse.
- **trg_prevent_multiple_managers:** Previne alocarea mai multor manageri pentru același restaurant.
- **trg_check_restaurant_capacity:** Verifică dacă adăugarea unei mese noi depășește capacitatea totală a restaurantului și previne această acțiune dacă este necesar.
- **trg_check_seating_capacity:** Acest trigger verifică dacă numărul total de oaspeți pentru o rezervare (incluzând clientul) nu depășește capacitatea mesei rezervate.

❖ FUNCȚII:

1. get_total_confirmed_reservation:

Scop: Returnează numărul total de rezervări confirmate.

Utilizare: Pentru raportarea și monitorizarea numărului de rezervări confirmate în sistem.

2. get_total_customers_and_guests_for_restaurant:

Scop: Returnează suma totală a clienților și invitaților pentru un anumit restaurant.

Utilizare: Pentru evaluarea și gestionarea capacității de locuri și a resurselor unui restaurant specific.



❖ PACHETE ȘI PROCEDURI :

Proceduri pentru inserare, actualizare și ștergere cu următoarele pachete:

1. **Pachet customers_pkg**
2. **Pachet reservations_pkg**
3. **Pachet restaurantmanagers_pkg**
4. **Pachet restaurants_pkg**
5. **Pachet tablesps_pkg**

Alte proceduri:

1. delete_customer_and_reservations

Scop: Această procedură asigură ștergerea unui client și a tuturor rezervărilor asociate acestuia.

2. ListReservationsForCustomer

Scop: Această procedură listează toate rezervările asociate unui client specificat.

Logica:

Definirea unui cursor (reservations_cursor) pentru selectarea rezervărilor din tabela reservations unde customer_id este egal cu p_customer_id.

Se parcurg toate rezervările folosind cursorul și se afișează detaliile fiecărei rezervări (reservation_id, reservation_date, number_of_guests, status, table_id).

În caz de eroare, se afișează un mesaj de eroare.

3. update_restaurant_capacity

Scop: Această procedură actualizează capacitatea totală a unui restaurant specificat.

4. UpdateReservation

Scop: Această procedură actualizează detaliile unei rezervări specificate sau o șterge dacă starea rezervării este "Cancelled".

Testele care pot fi rulate pentru a demonstra atingerea scopului propus

- Teste de Inserare
- Teste de Ștergere
- Teste de Actualizare
- Test de Capacitate:

Inserează o masă și rezervări pentru acea masă.

Încearcă să inserezi o rezervare care ar depăși capacitatea mesei și



verifică dacă triggerul `trg_check_seating_capacity` aruncă o eroare.

- **Test de Raportare:**

Apelează funcția `get_total_customers_and_guests_for_restaurant` și verifică dacă returnează numărul total corect de clienți și invitați pentru un restaurant specificat.

Concluzie

În concluzie, proiectul nostru a demonstrat că este posibil să se creeze un sistem robust și eficient de gestionare a rezervărilor pentru restaurante, care îmbunătățește semnificativ eficiența operațională și satisfacția clienților, asigurând în același timp integritatea și consistența datelor.