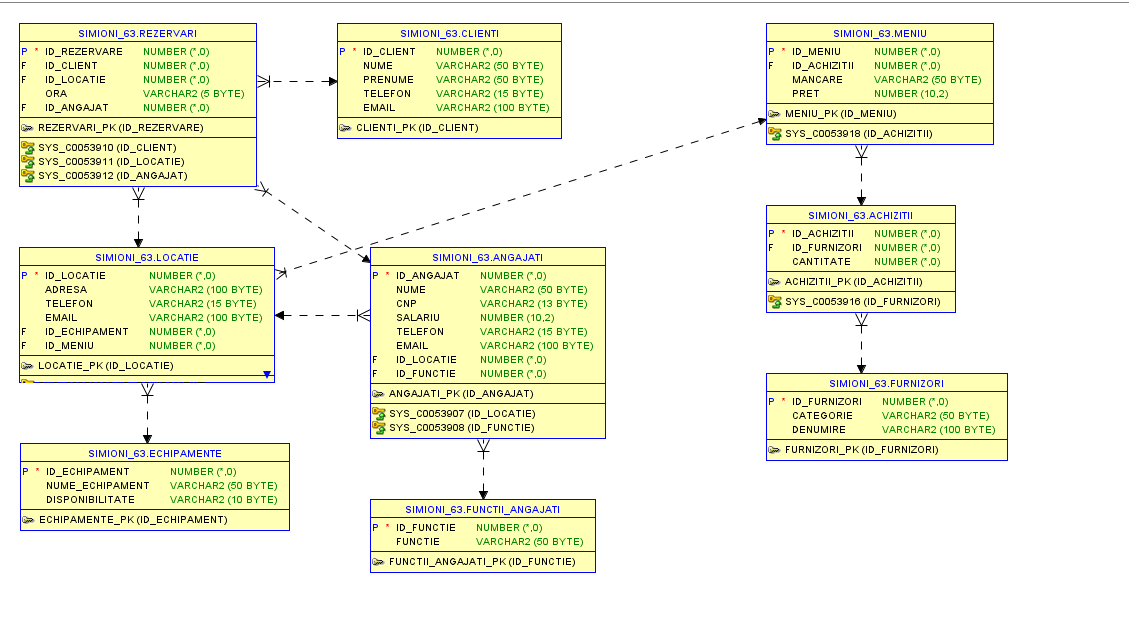
Aceasta baza de date este ca o agenda pentru un restaurant de lux. Ne gandim la ea ca la un organizator destept care tine evidenta clientilor, rezervarilor, angajatilor si a tot ce tine de functionarea restaurantului.

Asadar, avem lista de clienti, cu nume, prenume, telefon si email. Rezervarile sunt, practic, notite despre cine vine, unde si cand. Angajatii, adica oamenii din spatele culiselor, sunt inregistrati cu nume, CNP, salariu si alte detalii utile.

Exista si o lista de functii pentru angajati, precum ospatar, bucatar sau receptioner. Locatiile, adica restaurantele in sine, au detaliile lor, inclusiv echipamente si meniuri.

Deci, in esenta, aceasta baza de date face ca totul sa functioneze ca un ceas elvetian pentru restaurant, asigurandu-se ca, clientii primesc mancare buna si angajatii au tot ce le trebuie pentru a face asta posibil.



Crearea tabelului "Clienti":

CREATE TABLE Clienti (

ID\_Client INT PRIMARY KEY,

Nume VARCHAR(50),

Prenume VARCHAR(50),

Telefon VARCHAR(15),

Email VARCHAR(100)

);

Crearea tabelului "Rezervari":

CREATE TABLE Rezervari (

ID\_Rezervare INT PRIMARY KEY,

ID\_Client INT,

ID\_Locatie INT,

Ora VARCHAR(5),

ID\_Angajat INT,

FOREIGN KEY (ID\_Client) REFERENCES Clienti(ID\_Client),

FOREIGN KEY (ID\_Locatie) REFERENCES Locatie(ID\_Locatie),

FOREIGN KEY (ID\_Angajat) REFERENCES Angajati(ID\_Angajat)

);

Crearea tabelului "Angajati":

CREATE TABLE Angajati (

ID\_Angajat INT PRIMARY KEY,

Nume VARCHAR(50),

CNP VARCHAR(13),

Salariu DECIMAL(10, 2),

Telefon VARCHAR(15),

Email VARCHAR(100),

ID\_Locatie INT,

ID\_Functie INT,

FOREIGN KEY (ID\_Locatie) REFERENCES Locatie(ID\_Locatie),

FOREIGN KEY (ID\_Functie) REFERENCES Functii\_Angajati(ID\_Functie)

);

Crearea tabelului "Functii\_Angajati":

CREATE TABLE Functii\_Angajati (

ID\_Functie INT PRIMARY KEY,

Functie VARCHAR(50)

);

Crearea tabelului "Locatie":

CREATE TABLE Locatie (

ID\_Locatie INT PRIMARY KEY,

Adresa VARCHAR(100),

Telefon VARCHAR(15),

Email VARCHAR(100),

ID\_Echipament INT,

ID\_Meniu INT,

FOREIGN KEY (ID\_Echipament) REFERENCES Echipamente(ID\_Echipament),

FOREIGN KEY (ID\_Meniu) REFERENCES Meniu (ID\_Meniu)

);

Crearea tabelului "Echipamente":

CREATE TABLE Echipamente (

ID\_Echipament INT PRIMARY KEY,

Nume\_Echipament VARCHAR(50),

Disponibilitate VARCHAR(10)

);

Crearea tabelului "Meniu":

CREATE TABLE Meniu (

ID\_Meniu INT PRIMARY KEY,

ID\_Achizitii INT,

Mancare VARCHAR(50),

Pret DECIMAL(10, 2),

FOREIGN KEY (ID\_Achizitii) REFERENCES Achizitii(ID\_Achizitii)

);

Crearea tabelului "Achizitii":

CREATE TABLE Achizitii (

ID\_Achizitii INT PRIMARY KEY,

ID\_Furnizori INT,

Cantitate INT,

FOREIGN KEY (ID\_Furnizori) REFERENCES Furnizori(ID\_Furnizori)

);

Crearea tabelului "Furnizori":

CREATE TABLE Furnizori (

ID\_Furnizori INT PRIMARY KEY,

Categorie VARCHAR(50),

Denumire VARCHAR(100)

);

Inserarea datelor în tabelul "Clienti":

INSERT INTO Clienti (ID\_Client, Nume, Prenume, Telefon, Email)

VALUES

(1, 'Popescu', 'Ion', '123456789', 'ion.popescu@example.com');

INSERT INTO Clienti (ID\_Client, Nume, Prenume, Telefon, Email)

VALUES

(2, 'Ionescu', 'Ana', '987654321', 'ana.ionescu@example.com');

INSERT INTO Clienti (ID\_Client, Nume, Prenume, Telefon, Email)

VALUES

(3, 'Radu', 'Mihai', '555666777', 'mihai.radu@example.com');

Inserarea datelor în tabelul "Rezervari":

INSERT INTO Rezervari (ID\_Rezervare, ID\_Client, ID\_Locatie, Ora, ID\_Angajat)

VALUES

(1, 1, 1, '19:00', 101);

INSERT INTO Rezervari (ID\_Rezervare, ID\_Client, ID\_Locatie, Ora, ID\_Angajat)

VALUES

(2, 2, 2, '20:30', 102);

INSERT INTO Rezervari (ID\_Rezervare, ID\_Client, ID\_Locatie, Ora, ID\_Angajat)

VALUES

(3, 3, 1, '18:45', 103);

Inserarea datelor în tabelul "Angajati":

INSERT INTO Angajati (ID\_Angajat, Nume, CNP, Salariu, Telefon, Email, ID\_Locatie, ID\_Functie)

VALUES

(101, 'Popa', '1234567890123', 3000.00, '111222333', 'popa@example.com', 1, 1);

INSERT INTO Angajati (ID\_Angajat, Nume, CNP, Salariu, Telefon, Email, ID\_Locatie, ID\_Functie)

VALUES

(102, 'Georgescu', '9876543210987', 3500.00, '444555666', 'georgescu@example.com', 2, 2);

INSERT INTO Angajati (ID\_Angajat, Nume, CNP, Salariu, Telefon, Email, ID\_Locatie, ID\_Functie)

VALUES

(103, 'Stoica', '1111222233334', 2800.00, '777888999', 'stoica@example.com', 1, 3);

Inserarea datelor în tabelul "Functii\_Angajati":

INSERT INTO Functii\_Angajati (ID\_Functie, Functie)

VALUES

(1, 'Ospatar');

INSERT INTO Functii\_Angajati (ID\_Functie, Functie)

VALUES

(2, 'Bucatar');

INSERT INTO Functii\_Angajati (ID\_Functie, Functie)

VALUES

(3, 'Receptioner');

Inserarea datelor în tabelul "Locatie":

INSERT INTO Locatie (ID\_Locatie, Adresa, Telefon, Email, ID\_Echipament, ID\_Meniu)

VALUES

(1, 'Str. Restaurantului, nr. 123', '555111222', 'contact@example-restaurant.com', 1, 1);

INSERT INTO Locatie (ID\_Locatie, Adresa, Telefon, Email, ID\_Echipament, ID\_Meniu)

VALUES

(2, 'Bulevardul Culinar, nr. 45', '333444555', 'info@example-culinary.com', 2, 2);

Inserarea datelor în tabelul "Echipamente":

INSERT INTO Echipamente (ID\_Echipament, Nume\_Echipament, Disponibilitate)

VALUES

(1, 'Aparat de cafea', 'Da');

INSERT INTO Echipamente (ID\_Echipament, Nume\_Echipament, Disponibilitate)

VALUES

(2, 'Cuptor cu microunde', 'Nu');

Inserarea datelor în tabelul "Meniu":

INSERT INTO Meniu (ID\_Meniu, ID\_Achizitii, Mancare, Pret)

VALUES

(1, 1, 'Pizza Quattro Formaggi', 25.00);

INSERT INTO Meniu (ID\_Meniu, ID\_Achizitii, Mancare, Pret)

VALUES

(2, 2, 'File de somon cu sparanghel', 30.00);

INSERT INTO Meniu (ID\_Meniu, ID\_Achizitii, Mancare, Pret)

VALUES

(3, 3, 'Salata Caesar cu pui', 15.00);

Inserarea datelor în tabelul "Achizitii":

INSERT INTO Achizitii (ID\_Achizitii, ID\_Furnizori, Cantitate)

VALUES

(1, 1, 50);

INSERT INTO Achizitii (ID\_Achizitii, ID\_Furnizori, Cantitate)

VALUES

(2, 2, 20);

INSERT INTO Achizitii (ID\_Achizitii, ID\_Furnizori, Cantitate)

VALUES

(3, 3, 30);

Inserarea datelor în tabelul "Furnizori":

INSERT INTO Furnizori (ID\_Furnizori, Categorie, Denumire)

VALUES

(1, 'Alimente', 'Furnizor Alimente SRL');

INSERT INTO Furnizori (ID\_Furnizori, Categorie, Denumire)

VALUES

(2, 'Peste', 'Furnizor Peste Exotic SRL');

INSERT INTO Furnizori (ID\_Furnizori, Categorie, Denumire)

VALUES

(3, 'Salate', 'Furnizor Salate Proaspete SRL');

Operații de actualizare:

Actualizarea telefonului pentru clientul cu ID\_Client 2:

UPDATE Clienti

SET Telefon = '555111000'

WHERE ID\_Client = 2;

Actualizarea salariului pentru angajatul cu ID\_Angajat 101:

UPDATE Angajati

SET Salariu = 3200.00

WHERE ID\_Angajat = 101;

Actualizarea adresei pentru locația cu ID\_Locatie 1:

UPDATE Locatie

SET Adresa = 'Str. Nouă, nr. 456'

WHERE ID\_Locatie = 1;

Actualizarea disponibilității pentru echipamentul cu ID\_Echipament 2:

UPDATE Echipamente

SET Disponibilitate = 'Da'

WHERE ID\_Echipament = 2;

Actualizarea prețului pentru meniul cu ID\_Meniu 3:

UPDATE Meniu

SET Pret = 18.00

WHERE ID\_Meniu = 3;

Operații de interogare cu joncțiuni și funcții scalare:

Interogarea detaliilor rezervărilor cu numele clientului:

SELECT Rezervari.ID\_Rezervare, Clienti.Nume, Clienti.Prenume, Locatie.Adresa, Rezervari.Ora

FROM Rezervari

JOIN Clienti ON Rezervari.ID\_Client = Clienti.ID\_Client

JOIN Locatie ON Rezervari.ID\_Locatie = Locatie.ID\_Locatie;

Interogarea prețului mediu al meniurilor:

SELECT AVG(Meniu.Pret) AS Pret\_Mediu

FROM Meniu;

Interogarea sumei salariilor angajaților pe funcție:

SELECT Functii\_Angajati.Functie, SUM(Angajati.Salariu) AS Suma\_Salarii

FROM Angajati

JOIN Functii\_Angajati ON Angajati.ID\_Functie = Functii\_Angajati.ID\_Functie

GROUP BY Functii\_Angajati.Functie;

Interogarea numelui angajatului și salariului pentru cei cu salariul peste 3000:

SELECT Nume, Salariu

FROM Angajati

WHERE Salariu > 3000;

Interogarea detaliilor furnizorilor care furnizează alimente:

SELECT Furnizori.Denumire, Furnizori.Categorie, Achizitii.Cantitate

FROM Furnizori

JOIN Achizitii ON Furnizori.ID\_Furnizori = Achizitii.ID\_Furnizori

WHERE Furnizori.Categorie = 'Alimente';

Operații de interogare cu grupări ale datelor:

Interogarea numărului de rezervări pe locație:

SELECT Locatie.Adresa, COUNT(Rezervari.ID\_Rezervare) AS Numar\_Rezervari

FROM Rezervari

JOIN Locatie ON Rezervari.ID\_Locatie = Locatie.ID\_Locatie

GROUP BY Locatie.Adresa;

Interogarea sumei achizițiilor pentru fiecare furnizor:

SELECT Furnizori.Denumire, SUM(Achizitii.Cantitate) AS Suma\_Achizitii

FROM Furnizori

JOIN Achizitii ON Furnizori.ID\_Furnizori = Achizitii.ID\_Furnizori

GROUP BY Furnizori.Denumire;

Interogarea funcției cu cel mai mare salariu:

SELECT Functii\_Angajati.Functie, MAX(Angajati.Salariu) AS Salariu\_Maxim

FROM Angajati

JOIN Functii\_Angajati ON Angajati.ID\_Functie = Functii\_Angajati.ID\_Functie

GROUP BY Functii\_Angajati.Functie;

Interogarea mediei cantităților achiziționate pentru fiecare furnizor:

SELECT Furnizori.Denumire, AVG(Achizitii.Cantitate) AS Medie\_Cantitate

FROM Furnizori

JOIN Achizitii ON Furnizori.ID\_Furnizori = Achizitii.ID\_Furnizori

GROUP BY Furnizori.Denumire;

Interogarea numelui angajatului și sumei tuturor achizițiilor efectuate de clientul respectiv:

SELECT Clienti.Nume, SUM(Achizitii.Cantitate) AS Suma\_Achizitii

FROM Clienti

LEFT JOIN Rezervari ON Clienti.ID\_Client = Rezervari.ID\_Client

LEFT JOIN Achizitii ON Rezervari.ID\_Rezervare = Achizitii.ID\_Achizitii

GROUP BY Clienti.Nume;

-- Adăugarea unei coloane noi în tabelul "Clienti"

ALTER TABLE Clienti

ADD COLUMN Data\_Inregistrare DATE;

-- Schimbarea tipului de date pentru coloana "Telefon" în tabelul "Clienti"

ALTER TABLE Clienti

ALTER COLUMN Telefon VARCHAR(20);

ALTER TABLE Clienti

DROP COLUMN Email;

-- Ștergerea unui furnizor din tabelul "Furnizori"

DELETE FROM Furnizori

WHERE ID\_Furnizori = 2;

-- Actualizarea disponibilității echipamentului cu ID\_Echipament 1 folosind DECODE

UPDATE Echipamente

SET Disponibilitate = DECODE(Disponibilitate, 'Da', 'Nu', 'Da')

WHERE ID\_Echipament = 1;

-- Actualizarea funcției angajatului cu salariul peste 3000 folosind CASE

UPDATE Angajati

SET ID\_Functie = CASE WHEN Salariu > 3000 THEN 4 ELSE ID\_Functie END;

-- Ștergerea rezervării cu ID\_Rezervare 2

DELETE FROM Rezervari

WHERE ID\_Rezervare = 2;

-- Ștergerea unui locații din tabelul "Locatie"

DELETE FROM Locatie

WHERE ID\_Locatie = 2;

-- Unirea rezervărilor de la locația 1 și locația 2

SELECT \* FROM Rezervari

WHERE ID\_Locatie = 1

UNION

SELECT \* FROM Rezervari

WHERE ID\_Locatie = 2;

-- Intersectarea furnizorilor care furnizează alimente și peste

SELECT Denumire FROM Furnizori

WHERE Categorie = 'Alimente'

INTERSECT

SELECT Denumire FROM Furnizori

WHERE Categorie = 'Peste';

-- Diferența dintre achiziții și rezervări

SELECT ID\_Achizitii FROM Achizitii

MINUS

SELECT ID\_Achizitii FROM Rezervari;