



SAKARYA ÜNİVERSİTESİ

BSM 211 VERİTABANI YÖNETİM SİSTEMLERİ PROJE ÇALIŞMASI

SARE NUR KEYHİDAR – G231210012

sare.keyhidar@ogr.sakarya.edu.tr

<https://github.com/sar-3/RestaurantManagementSystem>

Proje Konusu:

Projenin Tanıtımı:

Restaurant Yönetim Sistemi, restoranların günlük işlerini düzenli ve verimli bir şekilde yönetmesini sağlayan kapsamlı bir uygulamadır. Sistem, masa yönetimi, sipariş ve rezervasyon işlemleri, çalışan ve malzeme takibi gibi temel süreçleri entegre ederek iş yükünü hafifletir ve insan hatalarını en aza indirir. Masa kapasiteleri, bölgeye göre masa kısıtlamaları ve durum güncellemeleri gibi özelliklerle masa yönetimi kolaylaştırılırken, siparişlerin detaylı takibi ve otomatik hesaplamalar sayesinde süreçler hızlanır.

Restaurant Yönetim Sistemi, çalışanların vardiya ve izin durumlarını etkin bir şekilde takip ederek işletme içi organizasyonu geliştirir. Yeni çalışan ekleme, silinen çalışanların bilgilerinin saklanması ve pozisyona özel kural uygulamaları ile çalışan yönetimi sistematik hale getirilir. Ayrıca, malzeme yönetimi ve yemek içeriklerinin kontrolü sayesinde stoklar düzenli olarak takip edilir ve eksik malzemeler kolayca güncellenir.

Bu sistem, restoranların hem operasyonel süreçlerini optimize eder hem de müşteri memnuniyetini artırmayı hedefler. Rezervasyon ve sipariş süreçlerinde çakışmaların önüne geçerken, doğru ve anlık bilgi sunarak yöneticilerin daha hızlı ve doğru kararlar almasına olanak tanır. Restaurant Yönetim Sistemi, iş akışını kolaylaştırarak işletmelere rekabet avantajı sağlar.

İş Kuralları

- Her kategori, benzersiz bir kategori adına sahip olmalıdır. Kategoriler, silinmeden önce bağlı oldukları yemeklerin referansları kaldırılmalıdır.
- Her masa, belirli bir kapasiteye sahip olmalıdır (yalnızca 2, 4, 5, 6 veya 8 kişi olabilir).
- Masanın bölge değeri yalnızca "iç mekan", "bahçe", veya "teras" olabilir. Masaların başlangıç durumu varsayılan olarak "boş" olmalı, durumu yalnızca "boş", "dolu" veya "rezerve" olarak değiştirilebilir.
- Her çalışanın benzersiz bir iletişim numarası olmalıdır. Pozisyon değeri yalnızca "garson", "aşçı" veya "personel" olabilir. Çalışanların başlangıç tarihi, mevcut tarihten önce bir değer olamaz.
- Her yemek benzersiz bir isim ile tanımlanır. Yemek fiyatları pozitif bir değer olmalıdır. Her yemek bir kategoriye atanmalıdır. Yemek silindiğinde, ilişkili kayıtları otomatik olarak silinmelidir.
- Her sipariş bir masa numarası ve çalışan numarası ile ilişkilendirilmelidir.
- Bir siparişin toplam tutar değeri sistem tarafından hesaplanır.
- Siparişler silindiğinde, ilişkili kayıtları da silinir.
- Siparişteki toplam tutar değeri otomatik olarak hesaplanır (miktar * fiyat).
- Sipariş silindiğinde ilişkili detaylar da otomatik olarak silinir.
- Her ödeme bir siparişe ilişkilendirilmelidir.
- Ödeme türü yalnızca "nakit" veya "kredi kartı" olabilir.
- Her rezervasyon bir masaya atanmalıdır. Rezervasyon tarihi yalnızca mevcut tarihten sonra bir değer olabilir. Bitiş saati, başlangıç saatinden sonra bir değer olmalıdır.
- Rezervasyon yapılırken masa durumu "rezerve" olarak güncellenir.
- Her vardiya bir çalışana atanmalıdır. Aynı çalışanın bir gün içinde birden fazla vardiyası olamaz. Vardiya başlangıç saati, bitiş saatinden önce olmalıdır. Çalışan silindiğinde vardiyaları da otomatik olarak silinir.
- Her vardiya-çalışan ilişkisi benzersiz olmalıdır. Aynı vardiyaya birden fazla çalışan atanabilir.
- Her izin talebi bir çalışana atanmalıdır. Çalışan silindiğinde izin talepleri de otomatik olarak silinir. Aynı çalışan ve izin talebi çifti yalnızca bir kez kaydedilebilir.
- Garsonlar yalnızca çalışan tablosunda yer alan çalışanlardan seçilebilir.
- Aşçılar yalnızca çalışan tablosunda yer alan çalışanlardan seçilebilir. Deneyim derecesi, yalnızca "yeni", "orta" veya "uzman" değerlerinden biri olabilir.
- Personeller yalnızca çalışan tablosunda yer alan çalışanlardan seçilebilir. Görev alanı yalnızca "temizlik personeli", "depo görevlisi" veya "bulaşıkçı" olabilir.
- Silinen çalışanların isim, soyisim, pozisyon, iletişim_numarasi ve baslangic_tarihi gibi bilgileri korunur.
- Çalışanın silindiği tarih silinme tarihi otomatik olarak kaydedilir.
- Her malzeme benzersiz bir malzeme adı ile tanımlanır. Malzeme silindiğinde ilişkili kayıtları otomatik olarak silinir.
- 8 kişilik masalardan maksimum 3 adet olmalıdır. 6 kişilik masalardan maksimum 5 adet olmalıdır. 5 kişilik masalardan maksimum 4 adet olmalıdır. 4 kişilik masalardan maksimum 7 adet olmalıdır. 2 kişilik masalardan maksimum 6 adet olmalıdır.
- Bahçede yalnızca 2 ve 4 kişilik masalar olabilir, toplamda bu masalardan 5 adet olmalıdır. Terasta yalnızca 7 masa olmalıdır, bunlar 2, 4, 5, 6, ve 8 kişilik olabilir. 8

kişilik masadan maksimum 1, 6 kişilik masadan maksimum 1, 5 kişilik masadan maksimum 2 adet bulunabilir. 2 kişilik ve 4 kişilik masalar herhangi bir sayıya sahip olabilir, fakat toplamda 7 masa sınırına dikkat edilmelidir. İç mekanda toplamda 13 masa olmalıdır ve bu masalar 2, 4, 5, 6 ve 8 kişilik olabilir.

- Siparişlerin toplam tutarı, sipariş detaylarından hesaplanarak sipariş tablosunda güncellenir.
- Sipariş detayları üzerinden hesaplanan toplam tutar, ödeme tutarını sipariş tablosunda günceller.
- Menü tablosundan yemek fiyatı alınarak yeni yemekler için fiyat bilgisi eklenir.
- Yeni bir çalışan eklenmeden önce, telefon numarasının başka bir çalışan tarafından kullanılmadığı kontrol edilir.
- Siparişin verildiği masanın kapasitesi, kişisel sayısından küçükse hata mesajı verilir.
- Çalışanın izinli olduğu günlerde vardiya kaydı eklenemez ve bir çalışan, aynı hafta içerisinde en fazla 5 vardiya alabilir.
- Vardiya saatleri belirli olup, bu saatlerde değişiklik yapılması engellenir.
- Çalışanın izinli olduğu tarihlerdeki vardiyaları silinir.
- Sipariş alabilmesi için, çalışan pozisyonunun "garson" olması gereklidir.
- Aynı masa ve zaman diliminde başka bir rezervasyon yapılması engellenir.
- Yeni bir garson eklenirse, garson tablosuna başlangıçta servis sayısı 0 olan bir veri eklenir.
- Yeni bir aşçı eklenirse, aşçı tablosuna başlangıçta "yeni" deneyim derecesi ile bir veri eklenir.
- Yeni bir personel eklenirse, personel tablosuna rastgele bir görev atanır (temizlik personeli, depo görevlisi veya bulaşıkçı).
- Garson tablosundaki servis sayısı, sipariş sayısına göre güncellenir.
- Yemek içeriğinde bulunan malzemeler kontrol edilerek, eksik malzemeler yeni olarak eklenir ve yemek-malzeme ilişkisi oluşturulur.
- Silinen çalışanın bilgileri, silinen çalışanlar tablosuna kaydedilir.
- Rezervasyonun zamanına göre masanın durumu güncellenir. Gelecek rezervasyonlar için masa "rezerve", devam eden rezervasyonlar için "dolmuş" olarak işaretlenir.
- Eğer masada başka aktif bir rezervasyon yoksa, masa "boş" olarak işaretlenir. Aksi takdirde, masa "rezerve" olarak kalır.

İLİŞKİSEL CEBİR

Kategori(kategori_id INT, kategori_adi VARCHAR(50), PRIMARY KEY(kategori_id))

Masa(masa_id INT, kapasite INT, bolge VARCHAR(50), durum VARCHAR(50), PRIMARY KEY(masa_id))

Çalışan(calisan_id INT, isim VARCHAR(100), soyisim VARCHAR(100), pozisyon VARCHAR(50), iletisim_numarasi VARCHAR(15), baslangic_tarihi DATE, PRIMARY KEY(calisan_id))

Yemek(yemek_id INT, isim VARCHAR(100), fiyat DECIMAL(10, 2), kategori_id INT, icerik TEXT, PRIMARY KEY(yemek_id), FOREIGN KEY(kategori_id) REFERENCES Kategori(kategori_id))

Sipariş(siparis_id INT, masa_id INT, calisan_id INT, toplam_tutar DECIMAL(10, 2), siparis_tarihi TIMESTAMP, PRIMARY KEY(siparis_id), FOREIGN KEY(masa_id) REFERENCES Masa(masa_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id))

SiparişDetay(siparis_detay_id INT, siparis_id INT, yemek_id INT, miktar INT, fiyat DECIMAL(10, 2), toplam_tutar DECIMAL(10, 2), PRIMARY KEY(siparis_detay_id), FOREIGN KEY(siparis_id) REFERENCES Sipariş(siparis_id), FOREIGN KEY(yemek_id) REFERENCES Yemek(yemek_id))

Ödeme(odeme_id INT, odeme_tutari DECIMAL(10, 2), odeme_turu VARCHAR(50), odeme_tarihi TIMESTAMP, siparis_id INT, PRIMARY KEY(odeme_id), FOREIGN KEY(siparis_id) REFERENCES Sipariş(siparis_id))

Rezervasyon(rezervasyon_id INT, musteri_adSoyad VARCHAR(100), musteri_telefon VARCHAR(15), masa_id INT, tarih DATE, baslangic_saat TIME, bitis_saat TIME, kisi_sayisi INT, PRIMARY KEY(rezervasyon_id), FOREIGN KEY(masa_id) REFERENCES Masa(masa_id))

Vardiya(vardiya_id INT, calisan_id INT, tarih DATE, baslangic_saati TIME, bitis_saati TIME, PRIMARY KEY(vardiya_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id))

İzinTalebi(izin_id INT, calisan_id INT, baslangic_tarihi DATE, bitis_tarihi DATE, aciklama TEXT, PRIMARY KEY(izin_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id))

Çalışanİzin(calisan_id INT, izin_id INT, PRIMARY KEY(calisan_id, izin_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id), FOREIGN KEY(izin_id) REFERENCES İzinTalebi(izin_id))

Garson(calisan_id INT, servis_sayisi INT, PK(calisan_id), FK(calisan_id) REFERENCES Çalışan(calisan_id))

Personel(calisan_id INT, gorev VARCHAR(100), PRIMARY KEY(calisan_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id))

Aşçı(calisan_id INT, deneyim_derecesi VARCHAR(20), PRIMARY KEY(calisan_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id))

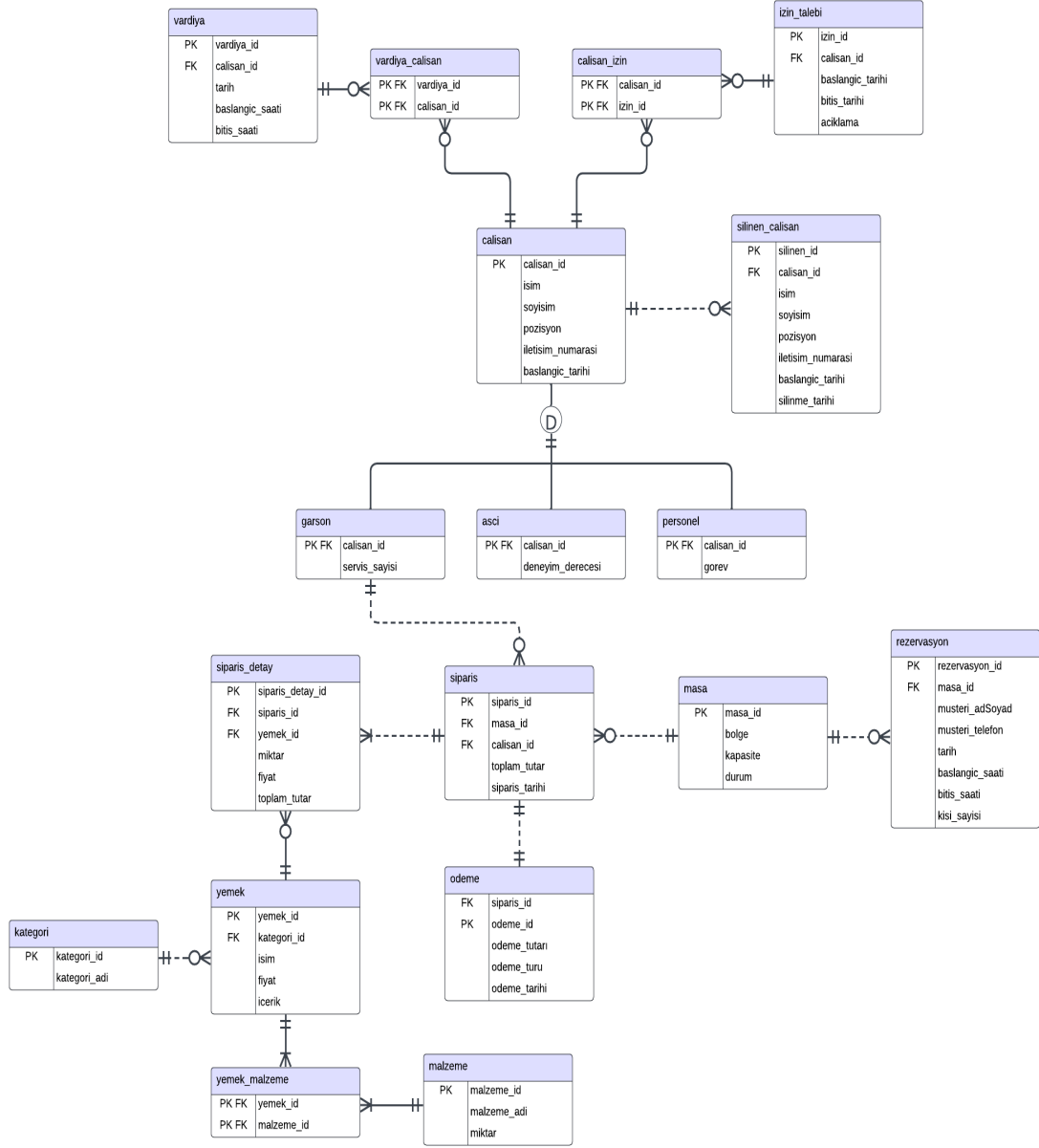
Malzeme(malzeme_id INT, malzeme_adi VARCHAR(100), miktar INT, PRIMARY KEY(malzeme_id))

YemekMalzeme(yemek_id INT, malzeme_id INT, PRIMARY KEY(yemek_id, malzeme_id), FOREIGN KEY(yemek_id) REFERENCES Yemek(yemek_id), FOREIGN KEY(malzeme_id) REFERENCES

Malzeme(malzeme_id)) VardiyaÇalışan(vardiya_id INT, calisan_id INT, PRIMARY KEY(vardiya_id, calisan_id), FOREIGN KEY(vardiya_id) REFERENCES Vardiya(vardiya_id), FOREIGN KEY(calisan_id) REFERENCES Çalışan(calisan_id))

SilinenÇalışan(silinen_id INT, calisan_id INT, isim VARCHAR(100), soyisim VARCHAR(100), pozisyon VARCHAR(50), iletisim_numarasi VARCHAR(15), baslangic_tarihi DATE, silinme_tarihi TIMESTAMP, PRIMARY KEY(silinen_id))

Crow's Foot



OLUŞTURDUĞUM TABLOLAR VE İÇERİŞİNE EKLEDİĞİM BİLGİLER

```

1  -- 1. Kategori Tablosu
2  ✓ CREATE TABLE IF NOT EXISTS kategori (
3      kategori_id SERIAL PRIMARY KEY,
4      kategori_adi VARCHAR(50) NOT NULL UNIQUE
5  );
6
7  -- 2. Masa Tablosu
8  ✓ CREATE TABLE IF NOT EXISTS masa (
9      masa_id SERIAL PRIMARY KEY,
10     kapasite INT NOT NULL CHECK (kapasite IN (2, 4, 5, 6, 8)),
11     bolge VARCHAR(50) NOT NULL CHECK (bolge IN ('iç mekan', 'bahçe', 'teraz')),
12     durum VARCHAR(50) CHECK (durum IN ('bos', 'dolü', 'rezerve')) DEFAULT 'bos'
13 );
14
15 -- 3. Çalışan Tablosu
16 ✓ CREATE TABLE IF NOT EXISTS calisan (
17     calisan_id SERIAL PRIMARY KEY,
18     isim VARCHAR(100) NOT NULL,
19     soyisim VARCHAR(100) NOT NULL,
20     pozisyon VARCHAR(50) NOT NULL CHECK (pozisyon IN ('garson', 'asci', 'personel')),
21     iletisim_numarasi VARCHAR(15) UNIQUE NOT NULL CHECK (iletisim_numarasi ~ '^\\+90[0-9]{10}$'),
22     baslangic_tarihi DATE NOT NULL DEFAULT CURRENT_DATE
23 );
24
25 -- 4. Yemek Tablosu
26 ✓ CREATE TABLE IF NOT EXISTS yemek (
27     yemek_id SERIAL PRIMARY KEY,
28     isim VARCHAR(100) UNIQUE NOT NULL,
29     fiyat DECIMAL(10, 2) NOT NULL,
30     kategori_id INT REFERENCES kategori(kategori_id),
31     icerik TEXT

```

```

CREATE INDEX idx_kategori_adi ON kategori(kategori_adi);
CREATE INDEX idx_yemek_kategori_id ON yemek(kategori_id);

```

```

37 -- 5. Sipariş Tablosu
38 ✓ CREATE TABLE IF NOT EXISTS siparis (
39     siparis_id SERIAL PRIMARY KEY,
40     masa_id INT NOT NULL REFERENCES masa(masa_id) ON DELETE CASCADE,
41     calisan_id INT NOT NULL REFERENCES calisan(calisan_id) ON DELETE SET NULL,
42     toplam_tutar DECIMAL(10, 2) DEFAULT 0,
43     siparis_tarihi TIMESTAMP DEFAULT CURRENT_TIMESTAMP
44 );
45
46 -- 6. Sipariş Detay Tablosu
47 ✓ CREATE TABLE IF NOT EXISTS siparis_detay (
48     siparis_detay_id SERIAL PRIMARY KEY,
49     siparis_id INT NOT NULL REFERENCES siparis(siparis_id) ON DELETE CASCADE,
50     yemek_id INT NOT NULL REFERENCES yemek(yemek_id) ON DELETE CASCADE,
51     miktar INT NOT NULL CHECK (miktar > 0),
52     fiyat DECIMAL(10, 2) DEFAULT 0,
53     toplam_tutar DECIMAL(10, 2) GENERATED ALWAYS AS (miktar * fiyat) STORED
54 );
55
56 CREATE INDEX idx_siparis_id ON siparis_detay(siparis_id);
57
58 -- 7. Ödeme Tablosu
59 ✓ CREATE TABLE IF NOT EXISTS odeme (
60     odeme_id SERIAL PRIMARY KEY,
61     odeme_tutari DECIMAL(10, 2) DEFAULT 0,
62     odeme_turu VARCHAR(50) CHECK (odeme_turu IN ('nakit', 'kredi karti')),
63     odeme_tarihi TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
64     siparis_id INT NOT NULL REFERENCES siparis(siparis_id) ON DELETE CASCADE
65 );
66
67 -- 8. Rezervasyon Tablosu
68 ✓ CREATE TABLE IF NOT EXISTS rezervasyon (
69     rezervasyon_id SERIAL PRIMARY KEY,
70     muster_i_adSoyad VARCHAR(100) NOT NULL,
71     muster_i_telefon VARCHAR(15) NOT NULL CHECK (muster_i_telefon ~ '^+90[0-9]{10}$'),
72     masa_id INT REFERENCES masa(masa_id) ON DELETE CASCADE,
73     tarih DATE NOT NULL CHECK (tarih BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '30 days'),
74     baslangic_saat TIME NOT NULL,
75     bitis_saat TIME NOT NULL CHECK (bitis_saat > baslangic_saat),
76     kisi_sayisi INT NOT NULL
77 );
78
79 -- 9. Vardiya Tablosu
80 ✓ CREATE TABLE IF NOT EXISTS vardiya (
81     vardiya_id SERIAL PRIMARY KEY,
82     calisan_id INT NOT NULL REFERENCES calisan(calisan_id) ON DELETE CASCADE,
83     tarih DATE NOT NULL,
84     baslangic_saati TIME NOT NULL DEFAULT '13:00:00',
85     bitis_saati TIME NOT NULL DEFAULT '22:00:00',
86     UNIQUE (calisan_id, tarih),
87     CHECK (baslangic_saati < bitis_saati)
88 );
89
90 -- 10. İzin Talebi Tablosu
91 ✓ CREATE TABLE IF NOT EXISTS izin_talebi (
92     izin_id SERIAL PRIMARY KEY,
93     calisan_id INT NOT NULL REFERENCES calisan(calisan_id) ON DELETE CASCADE,
94     baslangic_tarihi DATE NOT NULL,
95     bitis_tarihi DATE NOT NULL,
96     aciklama TEXT
97 );

```

```

99  --11.Çalışan ve İzin Talebi Tablosu
100 ✓ CREATE TABLE IF NOT EXISTS calisan_izin (
101     calisan_id INT NOT NULL REFERENCES calisan(calisan_id) ON DELETE CASCADE,
102     izin_id INT NOT NULL REFERENCES izin_talebi(izin_id) ON DELETE CASCADE,
103     PRIMARY KEY (calisan_id, izin_id)
104 );
105
106  -- 12. Garson Tablosu
107 ✓ CREATE TABLE garson (
108     calisan_id INT PRIMARY KEY REFERENCES calisan(calisan_id) ON DELETE CASCADE,
109     servis_sayisi INT DEFAULT 0
110 );
111
112  -- 13. Aşçı Tablosu
113 ✓ CREATE TABLE personel (
114     calisan_id INT PRIMARY KEY REFERENCES calisan(calisan_id) ON DELETE CASCADE,
115     gorev VARCHAR(100) CHECK (gorev IN ('temizlik personeli', 'depo görevlisi', 'bulaşıkçı'))
116 );
117
118  -- 14. Personel Tablosu
119 ✓ CREATE TABLE asci (
120     calisan_id INT PRIMARY KEY REFERENCES calisan(calisan_id) ON DELETE CASCADE,
121     deneyim_derecesi VARCHAR(20) CHECK (deneyim_derecesi IN ('yeni', 'orta', 'uzman')) DEFAULT 'yeni'
122 );
123
124  -- 15. Malzeme Tablosu
125 ✓ CREATE TABLE IF NOT EXISTS malzeme (
126     malzeme_id SERIAL PRIMARY KEY,
127     malzeme_adi VARCHAR(100) NOT NULL UNIQUE,
128     miktar INT NOT NULL CHECK (miktar >= 0)
129 );

```

```

131  --16. Yemek-Malzeme Tablosu
132 ✓ CREATE TABLE IF NOT EXISTS yemek_malzeme (
133     yemek_id INT REFERENCES yemek(yemek_id) ON DELETE CASCADE,
134     malzeme_id INT REFERENCES malzeme(malzeme_id) ON DELETE CASCADE,
135     PRIMARY KEY (yemek_id, malzeme_id)
136 );
137
138  -- 17. Vardiya-Çalışan Tablosu
139 ✓ CREATE TABLE IF NOT EXISTS vardiya_calisan (
140     vardiya_id INT NOT NULL REFERENCES vardiya(vardiya_id) ON DELETE CASCADE,
141     calisan_id INT NOT NULL REFERENCES calisan(calisan_id) ON DELETE CASCADE,
142     UNIQUE (vardiya_id, calisan_id)
143 );
144
145  -- 18. Silinen Çalışan Tablosu
146 ✓ CREATE TABLE IF NOT EXISTS silinen_calisan (
147     silinen_id SERIAL PRIMARY KEY,
148     calisan_id INT NOT NULL,
149     isim VARCHAR(100) NOT NULL,
150     soyisim VARCHAR(100) NOT NULL,
151     pozisyon VARCHAR(50) NOT NULL,
152     iletisim_numarasi VARCHAR(15),
153     baslangic_tarihi DATE,
154     silinme_tarihi TIMESTAMP DEFAULT CURRENT_TIMESTAMP
155 );
156

```



```

1504 ✓ INSERT INTO kategori (kategori_adi)
1505 VALUES
1506 ('Sıcak Başlangıç'),
1507 ('Soğuk Başlangıç'),
1508 ('Çorbalar'),
1509 ('Ana Yemekler'),
1510 ('Salatalar');
1511
1512 ✓ INSERT INTO yemek (isim, fiyat, kategori_id, icerik)
1513 VALUES
1514 ('Paçanga Böreği', 200.00, 1, 'yufka, domates, yeşil biber, kırmızı kapya biber, kaşar peyniri, pastırma'),
1515 ('Haşlama İçli Köfte', 180.00, 1, 'Kaburga eti, but eti, ince bulgur, süzme yoğurt'),
1516 ('Fındık Lahmacun', 90.00, 1, 'döş eti, biber, maydonoz, sarımsak'),
1517 ('Vişneli Hurmalı Sıcak Humus', 450.00, 1, 'Nohut, Konya Bozkır tahini, Ayvalık zeytinyağı, Limon, Vişne, Hurma'),
1518 ('Hatay Usulü Sıcak Humus', 400.00, 1, 'Nohut, Konya Bozkır tahini, Ayvalık zeytinyağı, Limon'),
1519 ('Zeytinyağlı Kuru Patlıcan Dolması', 170.00, 2, 'Antep patlıcanı, pirinç, maydonoz, sumak'),
1520 ('Vişneli Yaprak Sarma', 230.00, 2, 'Asma yaprağı, vişne, pirinç, kuru soğan, zeytinyağı, nane, dereotu, tuz, karabiber'),
1521 ('Soğan Dolması', 170.00, 2, 'soğan, kıyma, pirinç, domates salçası, zeytinyağı, tuz, karabiber, nane, maydanoz'),
1522 ('Şefin Zeytinyağlı Seçimi', 460.00, 2, 'tuz'),
1523 ('Mercimek Çorbası', 100.00, 3, 'kırmızı mercimek, kuru soğan, havuç, patates, un, zeytinyağı, tereyağı, tuz, karabiber, kırmızı toz'),
1524 ('Ezogelin Çorbası', 100.00, 3, 'kırmızı mercimek, bulgur, pirinç, kuru soğan, domates salçası, biber salçası, tereyağı, zeytinyağı, tuz, karabiber'),
1525 ('Terbiyeli Tavuk Çorbası', 150.00, 3, 'tavuk budu, şehriye, zeytinyağı, yoğurt, un, yumurta, sarımsak, Limon, karabiber, tuz');

```

```

1527 ✓ INSERT INTO masa (kapasite, bolge, durum)
1528 VALUES
1529 (2, 'iç mekan', 'bos'),
1530 (4, 'iç mekan', 'bos'),
1531 (4, 'bahçe', 'bos'),
1532 (6, 'iç mekan', 'bos'),
1533 (2, 'iç mekan', 'bos'),
1534 (6, 'teras', 'bos'),
1535 (8, 'teras', 'bos'),
1536 (2, 'iç mekan', 'bos'),
1537 (4, 'iç mekan', 'bos'),
1538 (4, 'bahçe', 'bos'),
1539 (6, 'iç mekan', 'bos'),
1540 (2, 'iç mekan', 'bos'),
1541 (6, 'iç mekan', 'bos'),
1542 (2, 'bahçe', 'bos'),
1543 (4, 'teras', 'bos');
1544
1545 ✓ INSERT INTO calisan (isim, soyisim, pozisyon, iletisim_numarasi, baslangic_tarihi)
1546 VALUES
1547 ('Emircan', 'Ağaç', 'asci', '+905345787345', '2024-12-21'),
1548 ('Zeynep', 'Bıçakcı', 'garson', '+904632472537', '2024-12-21'),
1549 ('Berat', 'Dere', 'personel', '+902438732687', '2024-12-21'),
1550 ('Azra', 'Balkaya', 'garson', '+904238746238', '2024-12-21'),
1551 ('Ceren', 'Albayrak', 'personel', '+903295743875', '2024-12-21'),
1552 ('Yusuf', 'Alemdaroğlu', 'asci', '+904938247938', '2024-12-21'),
1553 ('Esra', 'Nergis', 'garson', '+904324987583', '2024-12-21'),
1554 ('Bahar', 'Karagözoğlu', 'garson', '+905843579348', '2024-12-21'),
1555 ('Sude', 'Vergili', 'personel', '+909382759843', '2024-12-21'),
1556 ('Melisa', 'Çıkrıkçı', 'garson', '+904395873948', '2024-12-21'),

```

```

1560 ▾ INSERT INTO siparis (masa_id, calisan_id)
1561 VALUES
1562 (1, 2),
1563 (2, 2),
1564 (3, 4),
1565 (4, 4),
1566 (5, 8),
1567 (6, 7),
1568 (7, 7),
1569 (8, 8),
1570 (9, 12),
1571 (10, 10);
1572
1573 ▾ INSERT INTO malzeme (malzeme_adi, miktar)
1574 VALUES
1575 ('Kıyma', 120),
1576 ('Tavuk Göğsü', 100),
1577 ('Pirinç', 150);
1578
1579 ▾ INSERT INTO vardiya (calisan_id, tarih)
1580 VALUES
1581 (1, '2024-12-21'),
1582 (2, '2024-12-21'),
1583 (3, '2024-12-21'),
1584 (4, '2024-12-21'),
1585 (5, '2024-12-21'),
1586 (6, '2024-12-21'),
1587 (7, '2024-12-21'),
1588 (8, '2024-12-21'),
1589 (9, '2024-12-21'),

1592 ▾ INSERT INTO rezervasyon (musteri_adSoyad, muster_i_telefon, masa_id, tarih, baslangic_saat, bitis_saat, kisi_sayisi)
1593 VALUES
1594 ('Ahmet Yılmaz', '+905321234567', 1, '2024-12-28', '13:00:00', '15:00:00', 2),
1595 ('Elif Demir', '+905323456789', 2, '2024-12-23', '14:00:00', '16:00:00', 4),
1596 ('Fatma Kılıç', '+905324567890', 3, '2024-12-24', '15:00:00', '17:00:00', 3),
1597 ('Murat Çelik', '+905325678901', 4, '2024-12-25', '16:00:00', '18:00:00', 5),
1598 ('Zeynep Arslan', '+905326789012', 5, '2024-12-29', '17:00:00', '19:00:00', 2),
1599 ('Mehmet Gül', '+905327890123', 6, '2024-12-26', '18:00:00', '20:00:00', 6),
1600 ('Emine Kaya', '+905328901234', 7, '2024-12-27', '19:00:00', '21:00:00', 8),
1601 ('Ali Şahin', '+905329012345', 8, '2024-12-27', '20:00:00', '22:00:00', 2),
1602 ('Özge Aydın', '+905320123456', 11, '2024-12-24', '13:00:00', '15:00:00', 4),
1603 ('Gökhan Koç', '+905321234567', 12, '2024-12-25', '14:00:00', '16:00:00', 2),
1604 ('Ayşe Yıldız', '+905322345678', 13, '2024-12-23', '15:00:00', '17:00:00', 6),
1605 ('Canan Güngör', '+905323456789', 14, '2024-12-30', '16:00:00', '18:00:00', 2),
1606 ('Kemal Ekinci', '+905324567890', 15, '2024-12-25', '17:00:00', '19:00:00', 4),
1607 ('Nihan Bayraktar', '+905325678901', 9, '2024-12-28', '18:00:00', '20:00:00', 3),
1608 ('Baran Yılmaz', '+905326789012', 10, '2024-12-27', '19:00:00', '21:00:00', 4);

```

!!KODUN, FONKSİYONLARIN VE TRİGGERLARIN TAMAMI İLK SAYFADA BELİRTTİĞİM GİTHUB LİNKİNDE BULUNMAKTADIR.

(<https://github.com/sar-3/RestaurantManagementSystem/blob/main/RestaurantManagementSystem.sql>)

Saklı Yordam Örnekleri:

Kategori Ekleme Fonksiyonu:

Fonksiyonun adı `ekle_kategori` olup, amacı kategori tablosuna yeni bir kategori eklemektir. Bu fonksiyon bir `kategori_adi` parametresi alır ve verilen kategori adını tabloya ekler. Bunun için `INSERT INTO kategori(kategori_adi) VALUES (kategori_adi);` ifadesini kullanır. Bu işlem herhangi bir kontrol yapmadan yeni bir kategori ekler. Örneğin, `SELECT ekle_kategori('Elektronik');` ifadesi, 'Elektronik' adında bir kategori ekler.

Kategori Silme Fonksiyonu:

Fonksiyonun adı `sil_kategori` olup, amacı belirtilen `kategori_id` değerine sahip olan kategoriye kategori tablosundan silmektir. Fonksiyon, `p_kategori_id` adında bir parametre alır. Öncelikle bu ID'nin tabloda var olup olmadığını kontrol eder. Eğer kategori bulunamazsa, özel bir hata mesajı (`RAISE EXCEPTION`) döndürür. Kategori mevcutsa, `DELETE FROM kategori WHERE kategori_id = p_kategori_id;` ifadesiyle silme işlemini gerçekleştirir. Örneğin, `SELECT sil_kategori(5);` ifadesi, `kategori_id` değeri 5 olan kategoriye siler. Eğer böyle bir kategori yoksa, "Kategori ID 5 bulunamadı!" hatası döner.

```
159 -- Kategori Ekleme Fonksiyonu
160 CREATE OR REPLACE FUNCTION ekle_kategori(kategori_adi VARCHAR)
161 RETURNS VOID AS $$
162 BEGIN
163     INSERT INTO kategori(kategori_adi) VALUES (kategori_adi);
164 END;
165 $$ LANGUAGE plpgsql;
166
167 -- Kategori Silme Fonksiyonu
168 CREATE OR REPLACE FUNCTION sil_kategori(p_kategori_id INT)
169 RETURNS VOID AS $$
170 BEGIN
171     -- Kategoriye varlığını kontrol et
172     IF NOT EXISTS (SELECT 1 FROM kategori WHERE kategori_id = p_kategori_id) THEN
173         RAISE EXCEPTION 'Kategori ID % bulunamadı!', p_kategori_id;
174     ELSE
175         -- Kategori var ise sil
176         DELETE FROM kategori WHERE kategori_id = p_kategori_id;
177     END IF;
178 END;
179 $$ LANGUAGE plpgsql;
180
```

ekle_masa fonksiyonu, bir restoran yönetim sisteminde masa tablosuna yeni masalar eklemek için kullanılır ve ekleme işlemi sırasında belirli kontroller yapar. Fonksiyon, masanın kapasitesini (kapasite), bulunduğu bölgeyi (bolge) ve doluluk durumunu (durum) parametre olarak alır. Bölge yalnızca 'bahçe', 'teras', veya 'iç mekan' değerlerini, durum ise yalnızca 'boş' veya 'dolü' değerlerini kabul eder. Bu değerlerden herhangi biri geçersizse, özel bir hata mesajı döndürerek işlemi sonlandırır. Eğer tüm değerler geçerliyse, girilen parametreler kullanılarak masa tablosuna yeni bir kayıt ekler. Örneğin, `SELECT ekle_masa(4, 'bahçe', 'boş');` komutu, kapasitesi 4 olan, bahçede bulunan ve boş durumdaki bir masayı tabloya ekler.

```

197 -- Masa Ekleme Fonksiyonu
198 CREATE OR REPLACE FUNCTION ekle_masa(kapasite INTEGER, bolge TEXT, durum TEXT)
199 RETURNS VOID AS $$
200 BEGIN
201     -- Geçerli bölge kontrolü
202     IF bolge NOT IN ('bahçe', 'teras', 'iç mekan') THEN
203         RAISE EXCEPTION 'Geçersiz bölge: %! Lütfen "bahçe", "teras" veya "iç mekan" seçin.', bolge;
204     -- Geçerli durum kontrolü
205     ELSEIF durum NOT IN ('boş', 'dolü') THEN
206         RAISE EXCEPTION 'Geçersiz durum: %! Lütfen "boş" veya "dolü" seçin.', durum;
207     ELSE
208         -- Geçerli bölge ve durum ile masa ekleme
209         INSERT INTO masa (kapasite, bolge, durum)
210             VALUES (kapasite, bolge, durum);
211     END IF;
212 END;
213 $$ LANGUAGE plpgsql;
214 CREATE OR REPLACE FUNCTION ekle_masa(kapasite INTEGER, bolge TEXT, durum TEXT)
215 RETURNS VOID AS $$
216 BEGIN
217     -- Geçerli bölge kontrolü
218     IF bolge NOT IN ('bahçe', 'teras', 'iç mekan') THEN
219         RAISE EXCEPTION 'Geçersiz bölge: %! Lütfen "bahçe", "teras" veya "iç mekan" seçin.', bolge;
220     -- Geçerli durum kontrolü
221     ELSEIF durum NOT IN ('boş', 'dolü') THEN
222         RAISE EXCEPTION 'Geçersiz durum: %! Lütfen "boş" veya "dolü" seçin.', durum;
223     ELSE
224         -- Geçerli bölge ve durum ile masa ekleme
225         INSERT INTO masa (kapasite, bolge, durum)
226             VALUES (kapasite, bolge, durum);
227     END IF;
228 END;
229 $$ LANGUAGE plpgsql;
230

```

guncelle_vardiya fonksiyonu, bir vardiyanın başlangıç ve bitiş saatlerini güncellemek için kullanılır ve iki temel kontrol içerir. İlk olarak, verilen p_vardiya_id değerine sahip bir vardiyanın varlığını kontrol eder; eğer böyle bir vardiya yoksa, hata mesajı döndürür. Ardından, yeni başlangıç saati (p_yeni_baslangic_saati) ve bitiş saati (p_yeni_bitis_saati) arasında geçerlilik kontrolü yapar; eğer başlangıç saati bitiş saatine eşit veya ondan sonra ise, yine hata mesajı döner. Tüm kontroller geçildikten sonra, vardiya tablosunda belirtilen vardiya_id'ye ait kayıt, girilen yeni saat değerleriyle güncellenir. Örneğin, `SELECT guncelle_vardiya(3, '09:00', '17:00');` komutu, ID'si 3 olan vardiyanın saatlerini günceller.

```

701 -- Vardiya Güncelleme Fonksiyonu
702 CREATE OR REPLACE FUNCTION guncelle_vardiya(p_vardiya_id INT, p_yeni_baslangic_saati TIME, p_yeni_bitis_saati TIME)
703 RETURNS VOID AS $$
704 BEGIN
705     -- Geçerli vardiya ID'si kontrolü
706     IF NOT EXISTS (SELECT 1 FROM vardiya WHERE vardiya_id = p_vardiya_id) THEN
707         RAISE EXCEPTION 'Geçersiz vardiya_id: %', p_vardiya_id;
708     END IF;
709
710     -- Yeni başlangıç ve bitiş saati kontrolü
711     IF p_yeni_baslangic_saati >= p_yeni_bitis_saati THEN
712         RAISE EXCEPTION 'Yeni başlangıç saati bitiş saatinden önce olmalı: % - %', p_yeni_baslangic_saati, p_yeni_bitis_saati;
713     END IF;
714
715     -- Vardiya saatlerini güncelle
716     UPDATE vardiya
717     SET baslangic_saati = p_yeni_baslangic_saati, bitis_saati = p_yeni_bitis_saati
718     WHERE vardiya_id = p_vardiya_id;
719 END;
720 $$ LANGUAGE plpgsql;

```

TRİGGER ÖRNEKLERİ:

kontrol_vardiya_limiti tetikleyici fonksiyonu, çalışanların vardiyalarının izin durumlarına ve haftalık limitlere uygun olarak atanmasını sağlar. İlk olarak, çalışanın izinli olduğu günlerde (izin_talebi tablosuna göre) vardiya atanmasını engeller; eğer yeni vardiya tarihi izin tarih aralığına denk gelirse hata fırlatır. İkinci olarak, bir çalışanın bir haftada alabileceği vardiya sayısını 5 ile sınırlar; aynı hafta içinde mevcut vardiyalar 5 veya daha fazlaysa yeni vardiya eklenmesine izin vermez. Kontrolleri geçerse, yeni vardiya kaydını onaylayarak işlemi devam ettirir.

```
941 CREATE OR REPLACE FUNCTION kontrol_vardiya_limiti()
942 RETURNS TRIGGER AS $$
943 BEGIN
944     -- Çalışanın izinli olduğu günlerde vardiya yazılmasını engelle
945     IF EXISTS (
946         SELECT 1
947         FROM izin_talebi
948         WHERE calisan_id = NEW.calisan_id
949             AND NEW.tarih BETWEEN baslangic_tarihi AND bitis_tarihi
950     ) THEN
951         RAISE EXCEPTION 'Çalışan izinli olduğu günlerde vardiya yazılamaz.';
952     END IF;
953
954     -- Çalışanın aynı hafta içindeki vardiya sayısını kontrol et
955     IF (SELECT COUNT(*)
956         FROM vardiya
957         WHERE calisan_id = NEW.calisan_id
958             AND DATE_PART('year', tarih) = DATE_PART('year', NEW.tarih)
959             AND DATE_PART('week', tarih) = DATE_PART('week', NEW.tarih)) >= 5 THEN
960         RAISE EXCEPTION 'Bir çalışan haftada en fazla 5 vardiya alabilir.';
961     END IF;
962
963     RETURN NEW;
964 END;
965 $$ LANGUAGE plpgsql;
```

rezervasyon_cakisma_kontrolu tetikleyici fonksiyonu, bir masanın aynı tarih ve saat aralığında birden fazla rezervasyon yapılmasını engellemek için kullanılır. Fonksiyon, rezervasyon tablosunda yeni bir rezervasyon eklenirken veya güncellenirken, aynı masada belirtilen saat aralığında (OVERLAPS operatörü ile) çakışan başka bir rezervasyon olup olmadığını kontrol eder. Eğer böyle bir çakışma tespit edilirse, "Bu masa için belirtilen saatler arasında başka bir rezervasyon mevcut!" şeklinde bir hata fırlatarak işlemi engeller. Mevcut rezervasyon güncelleniyorsa, aynı rezervasyon ID'sini (rezervasyon_id) kontrol dışı bırakarak bu durumu dikkate alır. Kontrol başarılı olursa işlem devam eder.

```
1004 CREATE OR REPLACE FUNCTION rezervasyon_cakisma_kontrolu()
1005 RETURNS TRIGGER AS $$
1006 BEGIN
1007     -- Çakışma kontrolü
1008     IF EXISTS (
1009         SELECT 1
1010         FROM rezervasyon
1011         WHERE masa_id = NEW.masa_id
1012             AND tarih = NEW.tarih
1013             AND (
1014                 (NEW.baslangic_saat, NEW.bitis_saat) OVERLAPS (baslangic_saat, bitis_saat)
1015             )
1016             AND rezervasyon_id != NEW.rezervasyon_id -- Aynı rezervasyonu kontrol dışı bırak
1017     ) THEN
1018         RAISE EXCEPTION 'Bu masa için belirtilen saatler arasında başka bir rezervasyon mevcut!';
1019     END IF;
1020
1021     RETURN NEW;
1022 END;
1023 $$ LANGUAGE plpgsql;
```

garson_servis_sayisi_guncelle tetikleyici fonksiyonu, bir garsonun gerçekleştirdiği sipariş sayısına göre garson tablosundaki servis_sayisi alanını günceller. Fonksiyon, siparis tablosunda yeni bir sipariş eklendiğinde veya güncellendiğinde, ilgili garsonun (calisan_id ile eşleşen) toplam sipariş sayısını hesaplar ve bu değeri garson tablosundaki servis_sayisi alanına yazar. Böylece, her sipariş ekleme veya güncelleme işleminde garsonun toplam servis sayısı otomatik olarak güncellenir.

```
1070 CREATE OR REPLACE FUNCTION garson_servis_sayisi_guncelle()
1071 RETURNS TRIGGER AS $$
1072 BEGIN
1073     -- Garson tablosundaki servis sayısını güncelle
1074     UPDATE garson
1075     SET servis_sayisi = (
1076         SELECT COUNT(*) FROM siparis WHERE calisan_id = NEW.calisan_id
1077     )
1078     WHERE calisan_id = NEW.calisan_id;
1079
1080     RETURN NEW;
1081 END;
1082 $$ LANGUAGE plpgsql;
```

yemek_malzeme_ekle tetikleyici fonksiyonu, bir yemeğin içeriğinde belirtilen malzemeleri malzeme ve yemek_malzeme tablolarıyla ilişkilendirmek için kullanılır. Fonksiyon, NEW.icerik alanında virgülle ayrılmış malzeme isimlerini tek tek işler. Her bir malzeme için önce malzeme tablosunda varlığı kontrol edilir; malzeme yoksa yeni bir kayıt eklenir ve malzeme_id alınır. Ardından, mevcut veya yeni eklenen malzeme ile yemek arasında ilişki kurularak yemek_malzeme tablosuna bir kayıt eklenir. Bu işlem, yemeğin içeriğindeki tüm malzemeler için tekrar edilir ve tetikleme işlemi tamamlanır.

```
1084 CREATE OR REPLACE FUNCTION yemek_malzeme_ekle()
1085 RETURNS TRIGGER AS $$
1086 DECLARE
1087     v_malzeme_adi TEXT; -- PL/pgSQL değişkeni için yeni bir isim
1088     malzeme_exists INT;
1089     v_malzeme_id INT;
1090 BEGIN
1091     -- Yemek içeriğindeki malzemeleri virgülle ayır
1092     FOR v_malzeme_adi IN
1093         SELECT unnest(string_to_array(NEW.icerik, ',')) -- İçeriği virgülle ayır
1094     LOOP
1095         -- Malzemenin tablodan var olup olmadığını kontrol et
1096         SELECT malzeme_id INTO v_malzeme_id
1097         FROM malzeme
1098         WHERE malzeme.malzeme_adi = v_malzeme_adi
1099         LIMIT 1; -- sadece bir kayıt al
1100
1101         -- Eğer malzeme yoksa, yeni malzeme ekle
1102         IF NOT FOUND THEN
1103             INSERT INTO malzeme (malzeme_adi, miktar)
1104             VALUES (v_malzeme_adi, 0) RETURNING malzeme_id INTO v_malzeme_id;
1105         END IF;
1106
1107         -- Yemek ile malzemeyi ilişkilendir
1108         INSERT INTO yemek_malzeme (yemek_id, malzeme_id)
1109         VALUES (NEW.yemek_id, v_malzeme_id);
1110     END LOOP;
1111
1112     RETURN NEW;
1113 END;
1114 $$ LANGUAGE plpgsql;
```


EKLEME

(Yeni bir rezervasyon ekledim.)

Ali Dere

+905325789018

Kişi Sayısı: 4

Masa No: 11

28 Aralık 2024 Cumartesi

Başlangıç Saati: 18:00:00

Bitiş Saati: 20:00:00

Ekle

Güncelle

Sil

rezervasyon_i	musteri_adso	musteri_telef	masa_id	tarih	baslangic_s:
3	Fatma Kılıç	+905324567890	3	24.12.2024	15:00:00
4	Murat Çelik	+905325678901	4	25.12.2024	16:00:00
5	Zeynep Arslan	+905326789012	5	29.12.2024	17:00:00
6	Mehmet Gül	+905327890123	6	26.12.2024	18:00:00
7	Emine			27.12.2024	19:00:00
8	Ali Şahin			27.12.2024	20:00:00
9	Özge			24.12.2024	13:00:00
10	Gökhan			25.12.2024	14:00:00
11	Ayşe			23.12.2024	15:00:00
12	Canan			30.12.2024	16:00:00
13	Kemal			25.12.2024	17:00:00
14	Nihan Bayraktar	+905325678901	9	28.12.2024	18:00:00
15	Baran Yılmaz	+905326789012	10	27.12.2024	19:00:00
2	Elif Demir	+905323456789	2	23.12.2024	14:00:00
16	Baran Yılmaz	+905326789012	10	27.12.2024	15:00:00

Ali Dere

+905325789018

Kişi Sayısı: 4

Masa No: 11

28 Aralık 2024 Cumartesi

Başlangıç Saati: 18:00:00

Bitiş Saati: 20:00:00

Ekle

Güncelle

Sil

rezervasyon_i	musteri_adso	musteri_telef	masa_id	tarih	baslangic_s:
4	Murat Çelik	+905325678901	4	25.12.2024	16:00:00
5	Zeynep Arslan	+905326789012	5	29.12.2024	17:00:00
6	Mehmet Gül	+905327890123	6	26.12.2024	18:00:00
7	Emine Kaya	+905328901234	7	27.12.2024	19:00:00
8	Ali Şahin	+905329012345	8	27.12.2024	20:00:00
9	Özge Aydın	+905320123456	11	24.12.2024	13:00:00
10	Gökhan Koç	+905321234567	12	25.12.2024	14:00:00
11	Ayşe Yıldız	+905322345678	13	23.12.2024	15:00:00
12	Canan Güngör	+905323456789	14	30.12.2024	16:00:00
13	Kemal Ekinci	+905324567890	15	25.12.2024	17:00:00
14	Nihan Bayraktar	+905325678901	9	28.12.2024	18:00:00
15	Baran Yılmaz	+905326789012	10	27.12.2024	19:00:00
2	Elif Demir	+905323456789	2	23.12.2024	14:00:00
16	Baran Yılmaz	+905326789012	10	27.12.2024	15:00:00
18	Ali Dere	+905325789018	11	28.12.2024	18:00:00

(Kategoriye göre yemekleri aratma)

Yemek Listesi

Yemek Adı: Paçanga Böreği, Fiyat: ₺200,00, İçerik: yufka, domates, yeşil biber, kırmızı kapya biber, kaşar peyniri, pastırma

Yemek Adı: Haşlama İçli Köfte, Fiyat: ₺180,00, İçerik: Kaburga eti, but eti, ince bulgur, süzme yoğurt

Yemek Adı: Fındık Lahmacun, Fiyat: ₺90,00, İçerik: döş eti, biber, maydonoz, sarımsak

Yemek Adı: Vişneli Hummalı Sıcak Humus, Fiyat: ₺450,00, İçerik: Nohut, Konya Bozkır tahini, Ayvalık zeytinyağı, Limon, Vişne, Hurma

Yemek Adı: Hatay Usulü Sıcak Humus, Fiyat: ₺400,00, İçerik: Nohut, Konya Bozkır tahini, Ayvalık zeytinyağı, Limon

Tamam

Yemek ID	Yemek İsmi	Fiyat	Kategori	İçerik
1	Paçanga Böreği	200,00	1	yufka, do
2	Haşlama İçli K...	180,00	1	Kaburga
3	Fındık Lahmacun	90,00	1	döş eti, b
4	Vişneli Hummalı...	450,00	1	Nohut, K
5	Hatay Usulü Sıc...	400,00	1	Nohut, K
6	Zeytinyağlı Kur...	170,00	2	Antep pa
7	Visneli Yaprak ...	230,00	2	Asma ya

Ekle

Sil

Güncelle

sıcak başlangıç

Ara

(9 numaralı çalışanın bir vardiyasını sildim.)

Çalışan No Seçin...

23 Aralık 2024 Pazartesi

Ekle

Sil

Güncelle

	vardiya_id	calisan_id	tarih	baslangic_saat	bitis_saati
▶	1	1	21.12.2024	13:00:00	22:00:00
	2	2	21.12.2024	13:00:00	22:00:00
	3	3	21.12.2024	13:00:00	22:00:00
	4	4	21.12.2024	13:00:00	22:00:00
	5	5	21.12.2024	13:00:00	22:00:00
	6	6	21.12.2024	13:00:00	22:00:00
	7	7	21.12.2024	13:00:00	22:00:00
	8	8	21.12.2024	13:00:00	22:00:00
	10	10	21.12.2024	13:00:00	22:00:00
*					

GÜNCELLEME

(Siparişte, siparişi oluşturan garsonu güncelledim.)

Masa No: 5

Çalışan Bilgisi:

7-Esra Nergis

Bilgi

Sipariş başarıyla güncellendi!

Tamam

	siparis_id	masa_id	calisan_id	toplam_tutar	siparis_tarihi
	1	1	2	0,00	23.12.2024 03:02
	2	2	2	0,00	23.12.2024 03:02
	3	3	4	0,00	23.12.2024 03:02
	4	4	4	0,00	23.12.2024 03:02
	5	5	8	0,00	23.12.2024 03:02
	6	6	7	0,00	23.12.2024 03:02
	7	7	7	0,00	23.12.2024 03:02
	8	8	8	0,00	23.12.2024 03:02
	9	9	12	0,00	23.12.2024 03:02
	10	10	10	0,00	23.12.2024 03:02
	11	15	8	2160,00	23.12.2024 11:04
*					

Masa No: 5

Çalışan Bilgisi:

7-Esra Nergis

Ekle

Sil

Güncelle

Yemek Ekle

	siparis_id	masa_id	calisan_id	toplam_tutar	siparis_tarihi
	1	1	2	0,00	23.12.2024 03:02
	2	2	2	0,00	23.12.2024 03:02
	3	3	4	0,00	23.12.2024 03:02
	4	4	4	0,00	23.12.2024 03:02
	6	6	7	0,00	23.12.2024 03:02
	7	7	7	0,00	23.12.2024 03:02
	8	8	8	0,00	23.12.2024 03:02
	9	9	12	0,00	23.12.2024 03:02
	10	10	10	0,00	23.12.2024 03:02
	11	15	8	2160,00	23.12.2024 11:04
	5	5	7	0,00	23.12.2024 03:02
*					

!!UYGULAMA KODUNUN TAMAMI İLK SAYFADA BELİRTTİĞİM GİTHUB LİNKİNDE BULUNMAKTADIR.

(<https://github.com/sar-3/RestaurantManagementSystem/tree/main/Yeni%20klas%C3%B6r>)