

Database Project

2024

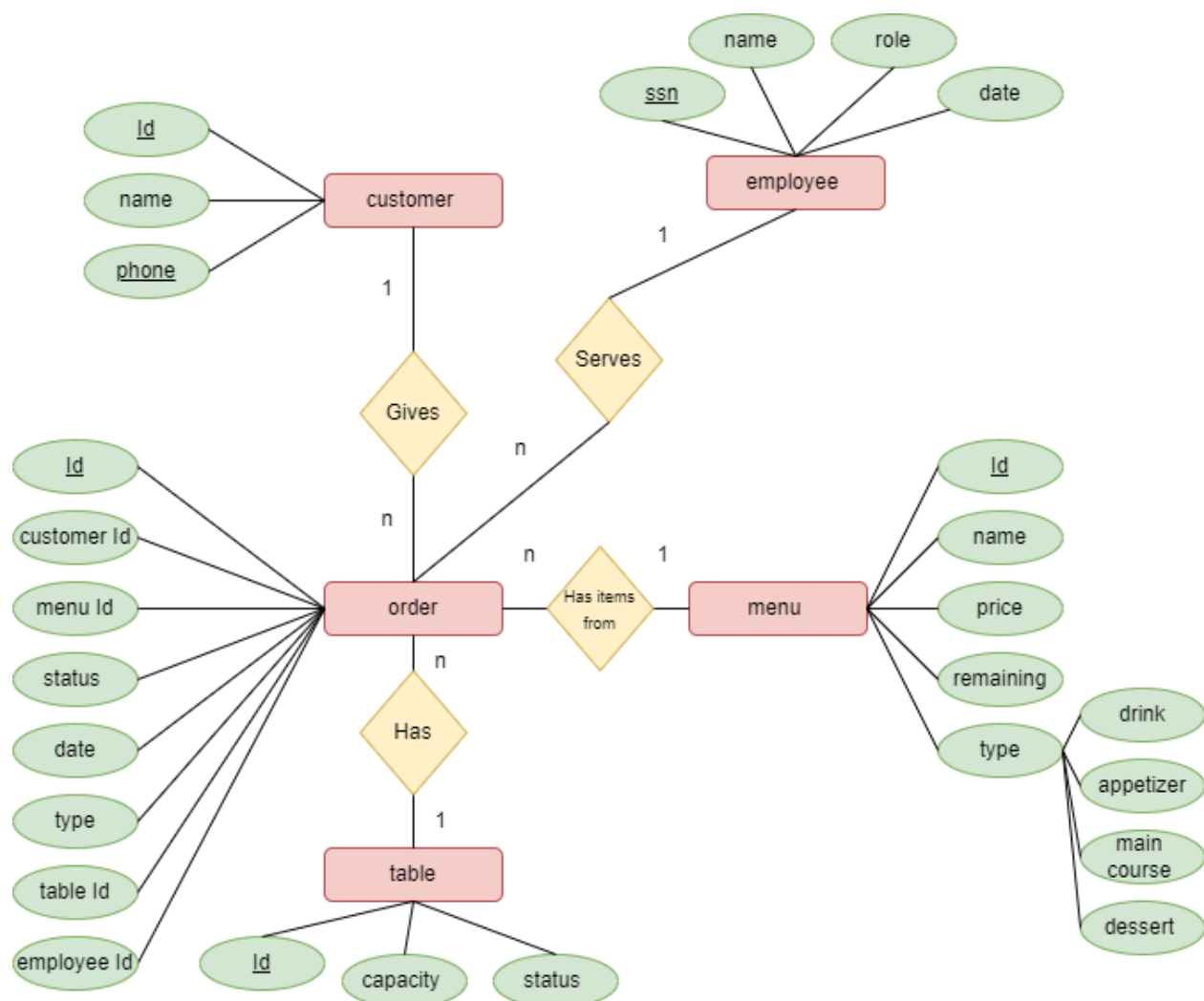
—

Shadmehr Salehi

Sina Asghari

Foozhan Fahimzade

ER



برای به دست آوردن شمای کلی از پروژه از جدول ER بالا استفاده کردیم.

SQL

در مرحله دوم برای ذخیره‌سازی اطلاعات Entity های موجود در ER را تبدیل به جدول‌های SQL کردیم.

1.Customer

	id	name	phone
1	1	Shadi	09301446373
2	2	Sina	09302224585
3	3	Foozhan	09124568735
4	4	Mahmoud	09128424587

شامل ستون‌های id, name, phone است که به ترتیب معرف شماره سطر، نام، شماره همراه است و phone به صورت یکتا مشتری مورد نظر را تعیین میکند.

2.Employee

	ssn	name	role	date
1	1112223344	David	Manager	2024-05-18
2	2223337788	Hassan	Waiter	2024-04-25
3	3334445566	Ali	Seller	2024-04-20

شامل ستون‌های ssn, name, role, date است که به ترتیب معرف شماره ملی، نام، عنوان شغلی، تاریخ استخدام است و ssn به صورت یکتا کارمند مورد نظر را تعیین میکند.

3.Menu

	id	name	type	remaining	price
1	1	Pizza	Main	4	50
2	2	Water	Drink	5	2
3	3	Ice Cream	Dessert	19	5
4	4	Pasta	Main	21	30
5	5	Burger	Main	9	20
6	7	Hotdog	Main	15	15

شامل ستون‌های id, name, type, remaining, price است که به ترتیب معرف شماره سطر، نام غذا، نوع غذا، تعداد باقی مانده، قیمت است که name به صورت یکتا غذای مورد نظر را تعیین میکند.

4.Table

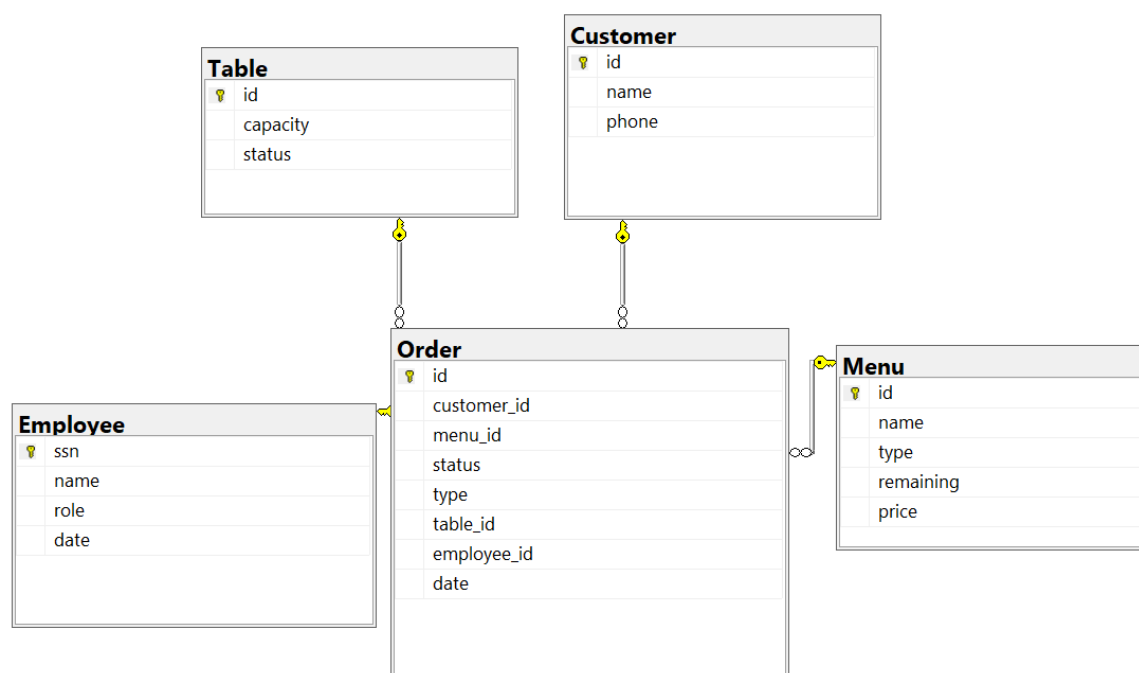
	id	capacity	status
1	1	4	empty
2	2	2	empty
3	3	4	empty
4	4	2	empty

شامل ستون‌های id, capacity, status است که به ترتیب معرف شماره میز، ظرفیت میز، وضعیت میز است که id به صورت یکتا میز مورد نظر را تعیین میکند.

5.Order

	id	customer_id	menu_id	status	type	table_id	employee_id	date
1	2	2	4	Pending	TakeAway	4	1112223344	2021-09-01
2	3	3	3	Done	TakeAway	1	1112223344	2024-04-10
3	5	1	1	Done	DineIn	1	1112223344	2024-05-01
4	6	2	1	Done	DineIn	4	1112223344	2024-05-01
5	7	3	4	Pending	DineIn	1	2223337788	2024-05-01
6	8	3	3	Done	DineIn	2	2223337788	2024-05-01

شامل ستون‌های id, customer_id, menu_id, status, type, table_id, date است که به ترتیب معرف شماره سفارش، مشتری سفارش دهنده، غذای سفارش داده شده، وضعیت سفارش، شماره میز، کارمند خدمت دهنده، تاریخ سفارش است که id به صورت یکتا سفارش مورد نظر را تعیین میکند. همچنین قابل توجه است که جدول موردنظر به چهار جدول Customer, menu, table, employee کلید خارجی دارد پس باید از داده‌های موجود در آن‌ها استفاده کند و به شکل زیر قابل نمایش است.



Implementation

در مرحله سوم به سراغ پیاده‌سازی frontend , backend و اتصال آن‌ها به database می‌رویم. برای اتصال به database از pyodbc استفاده شده است. برای هر جدول در یک کلاس جداگانه کد زده شده است و به صورت یک صفحه جداگانه نمایش داده می‌شود. همچنین کد مازوالر است بدین معنا که در صورتی که تصمیم به اضافه کردن یا حذف کردن یا تغییر دادن جدول گرفتیم بدون نیاز به آسیب زدن به بقیه جداول کد تغییر پیدا کند.

1.backend:

برای هر یک از جداول یک متد زده شده بدین صورت که ابتدا نوع عملیات را دریافت کرده و ستون‌های لازم برای جدول و عملیات مربوطه را دریافت می‌کند.

برای عملیات remove با استفاده از کلید اصلی داده را در جدول پیدا کرده و حذف می‌کند. برای عملیات update با استفاده از کلید اصلی داده را در جدول پیدا کرده و بقیه موارد داده شده را جایگزین می‌کند.

```
def modify_food(self,operation: Literal['add', 'remove','update'],
                name:str,
                Type:Optional[Literal['Main','Drink','Dessert']] = None,
                remaining:Optional[int]=None,price:Optional[int]=None) -> None:
```

سپس اطلاعات دریافت شده را به صورت یک رشته query درآورده و اجرا می‌کند.

```
try :
    query = f"INSERT INTO Menu (name,type,remaining,price) VALUES ('{name}','{Type}',{remaining},{price})"
    self.execute(query)
    print('food added successfully')
except Exception as e:
    print(f'Error inserting food \n error : {e}')
    return
```

همچنین یک متد برای نمایش جداول در نظر گرفته شده که اسم جدول را دریافت کرده و اطلاعات را برمیگرداند.

```
def show_tables(self, table: Literal['Menu', 'Customer', 'Employee', 'Order', 'Table']):  
    """  
    Show the content of a table in the database  
  
    Parameters  
    -----  
    table : str  
        | The name of the table to show {Menu, Customer, Employee, Order, Table}  
  
    Returns  
    -----  
    DataFrame  
        | The content of the table  
    """  
    query = f"SELECT * FROM [{table}]"  
    df = self.query(query)  
    df  
    return df
```

2.Frontend:

در این قسمت با استفاده از کتابخانه streamlit ستون‌های مورد نیاز را ایجاد کرده و اطلاعات را دریافت کرده و سپس متدهای نوشته شده در قسمت backend را فرا می‌خواند. برای نمایش اطلاعات از دکمه show استفاده میکنیم.

```
st.markdown("<h2 style='color: green;'>Menu management</h2>", unsafe_allow_html=True)

col1, col2, col3, col4 = st.columns(4)

with col1:
    Name = st.text_input("Name")

with col2:
    type = st.selectbox("type", ['Main', 'Drink', 'Dessert'])

with col3:
    remaining = st.number_input("remaining", value=0, step=1, format='%d')

with col4:
    price = st.number_input("price", value=0, step=1, format='%d')

if st.button("add"):
    db.modify_food('add', Name, type, remaining, price)

st.markdown("---")
```

Menu management

Name	type	remaining	price
<input type="text"/>	Main ▼	0 - +	0 - +

Log

ابتدا یک جدول برای لاگ‌ها ایجاد کردیم که شامل LogID, TableName, ChangeType, ChageTime, OriginalData, NewData است که به ترتیب معرف شماره لاگ، جدول تغییر یافته، نوع متد، زمان ایجاد لاگ، داده اولیه، داده تغییر یافته است.

	LogID	TableName	ChangeType	ChangeTime	OriginalData	NewData
1	1	Customer	INSERT	2024-06-17 19:45:52.627	NULL	[{"id":1021,"name":"MM","phone":"2225564 "}]
2	2	Customer	DELETE	2024-06-17 19:51:04.043	[{"id":1021,"name":"MM","phone":"2225564 "}]	NULL
3	3	Customer	UPDATE	2024-06-17 19:52:05.753	[{"id":1018,"name":"Dorsaa","phone":"0925604072 ...	[{"id":1018,"name":"Dorsaa","phone":"0925604073 ...

برای خوانایی بهتر و واضح شدن تغییرات اعمال شده ستون changes در backend به جدول اضافه شده است که در صفحه وبسایت قابل مشاهده است.

	LogID	ChangeTime	TableName	ChangeType	OriginalData	NewData	Changes
0	1	2024-06-17 19:45:52	Customer	INSERT	None	[{"id":1021,"name":"MM","phone":"2225564 "}]	Data Added
1	2	2024-06-17 19:51:04	Customer	DELETE	[{"id":1021,"name":"MM","phone":"2225564 "}]	None	Data Deleted
2	3	2024-06-17 19:52:05	Customer	UPDATE	[{"id":1018,"name":"Dorsaa","phone":"0925604072 "}]	[{"id":1018,"name":"Dorsaa","phone":"0925604073 "}]	'phone': 0925604072 -> 0925604073
3	4	2024-06-18 17:02:15	Customer	INSERT	None	[{"id":1022,"name":"Ahmed","phone":"123456789 "}]	Data Added

میتوان با استفاده از زمان، نوع جدول، یا عمل انجام شده لاگ‌های مورد نظر در آن محدود را پیدا کرد.

Date

2024-06-17

Show

	LogID	ChangeTime	TableName	ChangeType	OriginalData
0	1	2024-06-17 19:45:52	Customer	INSERT	None
1	2	2024-06-17 19:51:04	Customer	DELETE	{"id":1021,"name":"MM","phone":"2225564 "}
2	3	2024-06-17 19:52:05	Customer	UPDATE	{"id":1018,"name":"Dorsaa","phone":"0925604072

Which Part?

Customer

Show_res

Which Action?

Insert

Show_res2