



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه پایانی

آزمایشگاه پایگاه داده

نگارش  
آرش حاجی صفی - ۹۶۳۱۰۱۹

بهمن ۱۳۹۹

## توضیحات پروژه:

هدف این پروژه طراحی دیتابیس مربوط به یک رستوران می‌باشد. در این سامانه، منوی رستوران و فاکتور سفارش به صورت الکترونیکی ذخیره می‌شود. مشتریان می‌توانند در رستوران ثبت‌نام نمایند و فاکتور به اسم آنها صادر شود؛ همچنین افراد می‌توانند به صورت مهمان سفارش ثبت کنند و فاکتور بدون اسم برایشان صادر شود. امکان ارسال سفارش به بیرون رستوران هم از طریق دریافت آدرس کاربر وجود دارد که برای این سفارش‌ها آدرس در فاکتور ظاهر می‌شود. همچنین امکان تغییر قیمت غذاهای موجود در منوی رستوران با گذر زمان وجود دارد.

در ضمن رستوران مواد اولیه خود را از فروشگاه‌های متفاوت خریداری می‌کند. بنابراین جداولی برای فروشگاه‌های اولیه و لیست اقلام موجود در آنها و قیمت هر کدام از آنها وجود دارد. وقتی رستوران از این فروشگاه‌ها خرید می‌کند، برایش فاکتوری صادر شود که نوع اقلام، تعداد و قیمت هر کدام و قیمت مجموع را در بر می‌گیرد. این اقلام موجود در فروشگاه‌ها هم در گذر زمان تغییر می‌کنند و قیمت آنها ممکن است دچار افزایش و یا کاهش شود.

تعامل کاربران به این صورت خواهد بود که منوی غذای رستوران برای هر روز محدود است و وقتی کاربری غذایی را سفارش می‌دهد از موجودی آن غذا کم می‌شود و ممکن است آن غذا ناموجود شود و کاربر دیگری نتواند آنرا سفارش دهد. همچنین برای هر کاربر یک کیف پول آنلاین در نظر گرفته شده است که بتواند سفارش خود را از موجودی آن پرداخت نمایند و یا به یکدیگر مبلغی را انتقال دهند.

برای رستوران ۴ نقش مختلف در نظر گرفته شده است؛ مدیر رستوران به اطلاعات جداول رستوران خود دسترسی دارد و می‌تواند خریدهای روزانه و فروش و سود هر روز را در قالب گزارش دریافت کند. مدیر فروشگاه‌ها هم به همین صورت به اطلاعات فروشگاه خود دسترسی دارند. ادمین سیستم به اطلاعات و گزارش‌های همه دسترسی دارد. کاربران نیز می‌توانند مجموع خریدهای خود از رستوران را ببینند و گزارش بگیرند که مثلاً کدام غذاها را بیشتر از همه سفارش داده‌اند.

همچنین به وسیله trigger لاگ‌های جداول را نگه می‌داریم و شماره تلفن را بررسی می‌کنیم. ۳ تا procedure داریم که یکی از آنها برای حذف لاگ‌هایی است که بیش از ۳ روز از طول عمر آنها می‌گذرد؛ یک procedure دیگر برای انتقال پول از یک کاربر به کاربر دیگر است، procedure آخر نیز برای اضافه کردن یک مبلغ مورد نظر به حساب همه کاربران است.

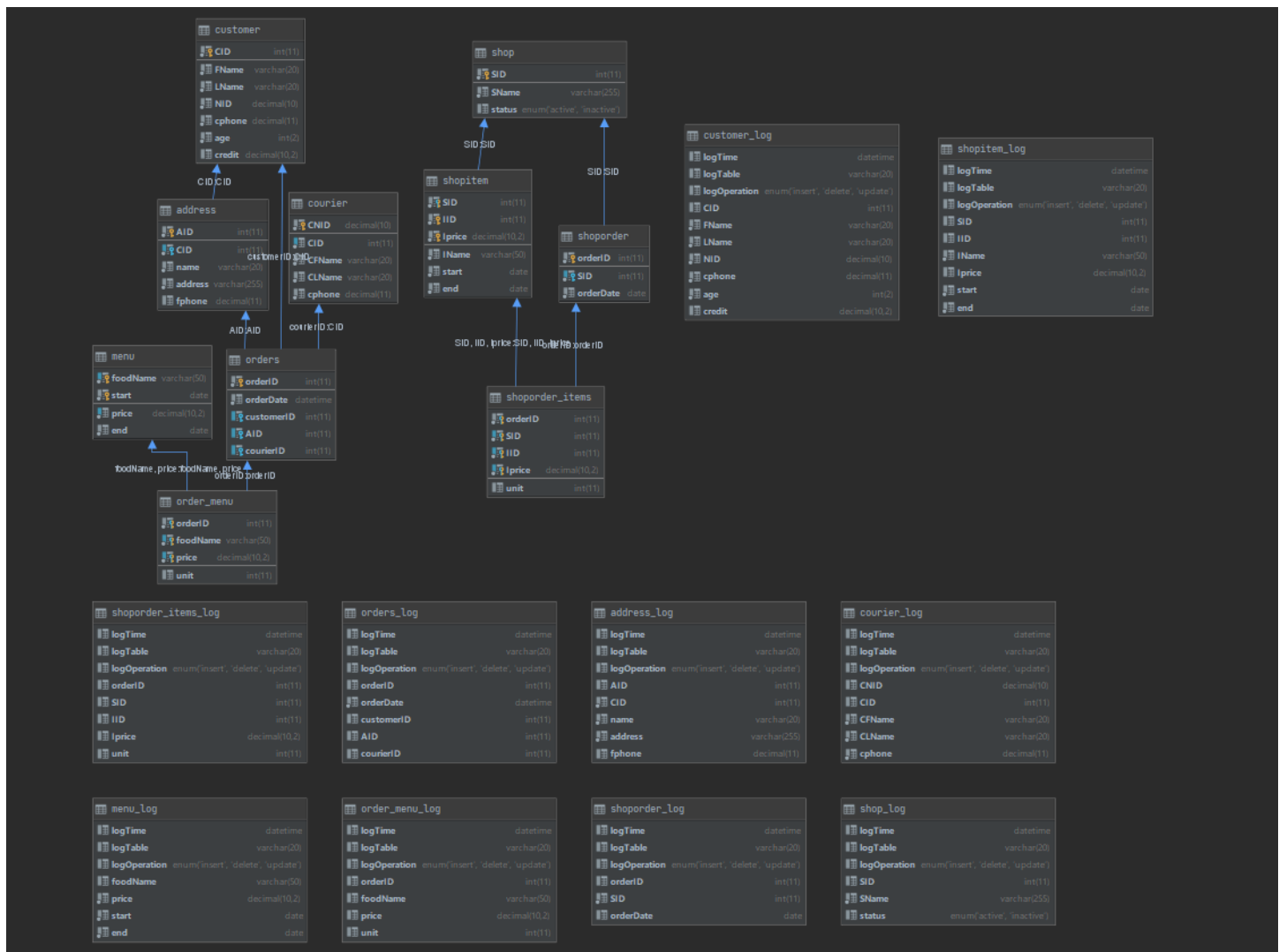
۳ تا function هم داریم؛ یکی برای بدست آوردن تعداد همه‌ی آدرس‌های موجود از یک کاربر، دیگری برای تعداد همه‌ی غذاهای موجود در منو، و آخری برای دریافت مجموع هزینه همه غذاهای منو.

این پروژه با پایگاه داده MySQL در زبان Python (با استفاده از درایور MySQL Connector) پیاده‌سازی شده است.

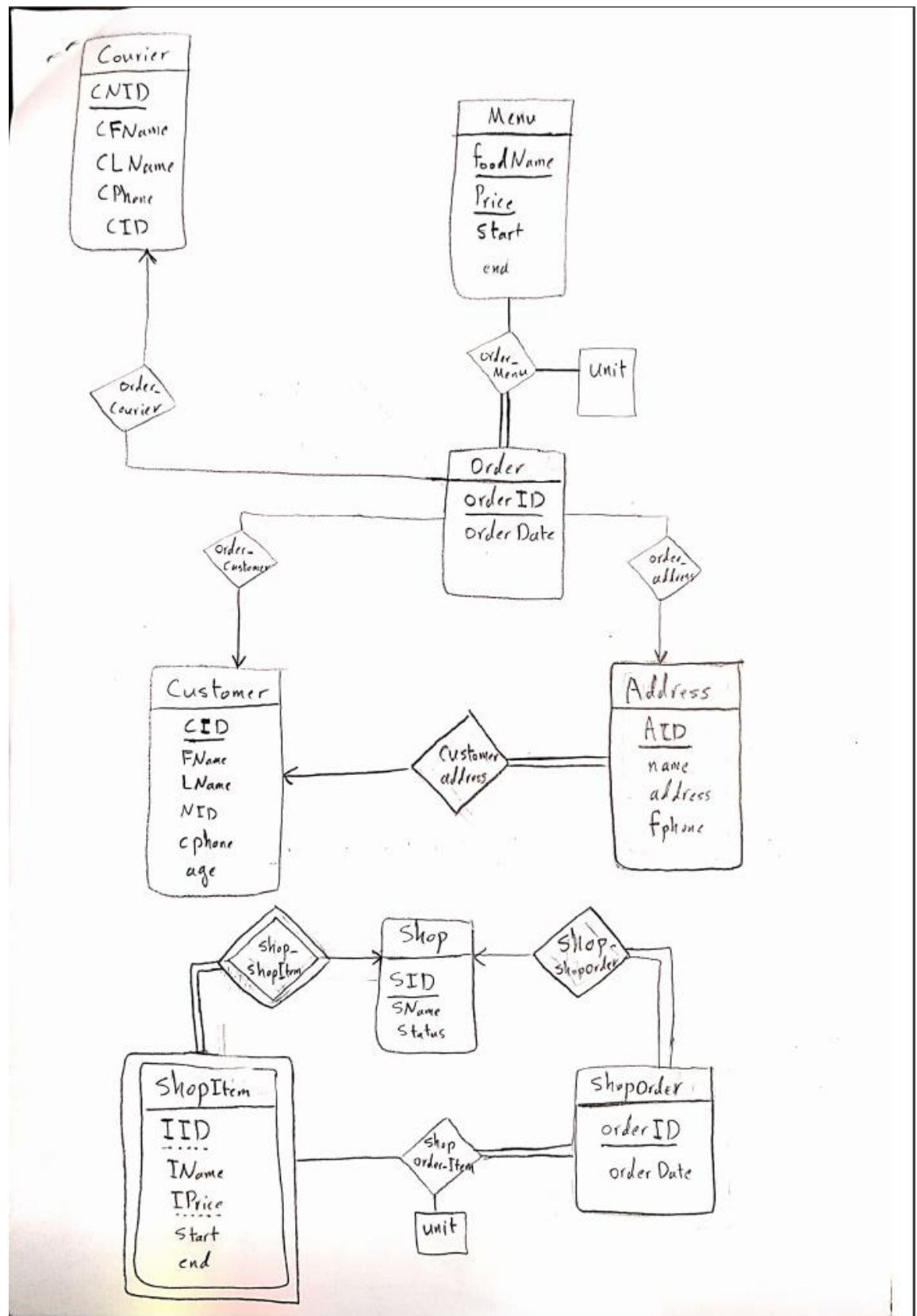
## طراحی پایگاه داده و حل چالش‌ها :

در عمل برای طراحی جداول از همان ERD که رسم کردم استفاده کردم به این صورت که روابط چند به یک را که قابل حذف شدن هستند، مانند رابطه‌ی بین سفارش و مشتری، حذف کرده و کلید قسمت یک را در قسمت چند قرار داده‌ام. این کار باعث بهبود کارایی جداول و کمتر شدن تعداد آنها و در نتیجه کنترل بهتر و ساده‌تر دیتابیس با هزینه‌ای بسیار کم شده است.

جداول نهایی (بدون در نظر گرفتن logها) به تعداد ۱۰ تا هستند که به صورت زیر قرار گرفته اند:



همچنین نمودار کامل ER این دیتابیس به صورت زیر است:



۱- جدول customer اطلاعات مشتری را در برمی گیرد.

۲- یکی از چالش‌ها multivalued بودن آدرس بود که در نتیجه جدول جدایی را برای آدرس گرفته ام که آیدی جدول مشتری در آنها قرار گرفته و هر آدرس متعلق به یک مشتری می‌شود و هر مشتری چندین آدرس می‌تواند داشته باشد.

۳- جدول courier اطلاعات پیک را در بر می‌گیرد، هر پیک هم به وسیله‌ی شماره ملی و هم آیدی می‌تواند به طور یونیک مشخص شود.

۴- هر رکورد جدول منو شامل اسم غذا، قیمت آن، تاریخ شروع و پایان است. یکی دیگر از چالش‌ها قابل تغییر بودن اسم و قیمت آیتم‌های منو بود به طوری که در فاکتور تغییری نکنند؛ برای حل این چالش یک تاریخ شروع و پایان گرفته ام که به این معنی است که قیمت هر غذا (یا حتی اسم آن) یک Temporary Variable است و تاریخ شروع و پایان، بازه‌ی valid بودن آنرا مشخص می‌کند. به این ترتیب لازم نیست دیگر رکوردی از منو حذف شود و وقتی قیمت آیتمی یا اسم آن تغییر می‌کند، رکورد قبلی invalid شده و رکورد جدید با تاریخ‌های جدید وارد منو می‌شود. در نتیجه برای رکوردهای فاکتورها مشکلی پیش نمی‌آید و دقیقاً همان اطلاعات اصلی آنها باقی می‌ماند. کلید اصلی هم برای منو، اسم غذا به همراه تاریخ شروع valid بودن آن می‌شود که البته اسم غذا به همراه قیمت را هم می‌توان کلید کاندید برای ادامه‌ی پروژه در نظر گرفت.

۵- جدول orders، آیدی سفارش (آیدی فاکتور) و تاریخ ثبت آنرا در بر می‌گیرد. مطابق تعریف پروژه هر رکورد این جدول همچنین می‌تواند با customerID به یک مشتری مرتبط شود، با AID مربوط به یک آدرس شود (که یعنی غذا بیرون بر است)، و با courierID به یک پیک مرتبط شود. این ۳ فیلد الزامی نیستند و می‌توانند null باشند.

۶- جدول order\_menu ارتباط بین سفارش با منو را برقرار می‌کند که یکی دیگر از چالش‌های پروژه بوده؛ یعنی مشخص می‌کند که هر سفارش شامل چه آیتم‌هایی از منو بوده و از هر نوع غذای موجود چند عدد در آن سفارش بوده. قیمت و اسم غذای مربوطه حتی با تغییر منو و invalid شدن رکوردهای قبلی هم همانطور که گفته شد عوض نمی‌شود و اطلاعات اصلی حفظ می‌شود. بنابراین چالش multivalued بودن غذاهای مربوط به هر سفارش هم در این جدول حل می‌شود.

۷- جدول shop اطلاعات هر فروشگاه و وضعیت آنرا نگه می‌دارد.

۸- جدول shoporder سفارش های رستوران از فروشگاه‌ها و تاریخ انجام سفارش را مشخص می‌کند. منطق این قسمت نزدیک به orders خود رستوران است.

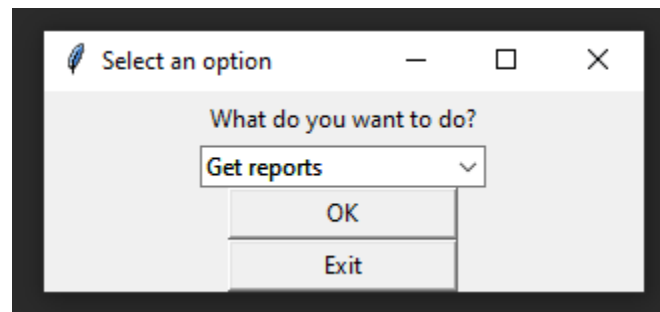
۹- جدول shopitem، آیتم‌های هر فروشگاه را در بر دارد. یکی دیگر از چالش‌ها نگهداری آیتم‌های فروشگاه‌های مختلف بود. چون آیتم فروشگاه یک weak entity است، این جدول کلید shop را هم در بر دارد. با این وجون باز هم چون قیمت و یا اسم آیتم‌های هر فروشگاه در گذر زمان تغییر می‌کنند، نمی‌توان یک رکورد را به طور یونیک مشخص کرد و چالش بعدی پیش می‌آید که در فاکتورهای خرید رستوران، این قیمت‌ها و اسم‌ها نباید تغییر کنند. برای این منظور مشابه منوی رستوران، تاریخ شروع و پایان valid بودن اطلاعات یک آیتم را هم نگه می‌داریم تا با گذر زمان و تغییر قیمت‌ها و یا اسم آیتم‌ها، تغییری در فاکتورهای خرید از فروشگاه‌ها به وجود نیاید.

۱۰- جدول shoporder\_items هر فاکتور یا خرید از فروشگاه توسط رستوران را به آیتم‌ها و قیمت آن فروشگاه متصل می‌کند. همانطور که توضیح داده شد چالش مربوط به تغییر نکردن اطلاعات فاکتور در گذر زمان و با تغییر قیمت آیتم‌ها یا غیر فعال شدن فروشگاه حل شده است.

به ازای هر جدول یک جدول هم برای log به وجود آمده که دارای ۳ فیلد (نام جدول، نوع عملیات، تاریخ عملیات) به علاوه تمامی فیلدهای جدول اصلی است که با trigger تمامی عملیات‌های insert/update/delete به همراه رکوردی که تغییر کرده در آن قرار می‌گیرد. پس مجموعاً ۱۰ تا جدول اصلی ایجاد شده است به علاوه ۱۰ تا جدول برای log. برای بررسی صحت شماره موبایل و تلفن ثابت در جدول‌های address و customer و courier هر کدام یک trigger نوشته شده که برای بررسی شماره موبایل چک می‌شود که حتماً ۱۰ رقم باشد (بدون صفر اول) و به فرم 9xxxxxxxx باشد و برای تلفن ثابت هم بررسی می‌شود که ۱۰ رقم (بدون صفر اول) باشد و در هر دو نوع شماره، هیچ کاراکتری قرار نگرفته باشد، وگرنه اجازه‌ی insert داده نمی‌شود.

یک stored procedure هم با نام refresh\_logs() ایجاد شده که هیچ پارامتر ورودی ندارد و در تمامی جداول log، اگر از تاریخ ایجاد رکوردی بیشتر از ۳ روز (نسبت به تاریخ فعلی) گذشته باشد، آن رکوردها را پاک می‌کند.

تمامی queryهای گزارش گیری ها هم از طریق منوی اول برنامه قابل اجرا هستند و نتیجهی گزارش هم در ترمینال پرینت شده و هم در فایل report.txt در فایلی که کد برنامه است، ذخیره می شود:



report.txt - Notepad

File Edit Format View Help

Total Sales

foodName	total_sold	unit_price	on_date
b	1	33	2020-01-30
a	4	30	2020-01-30
d	4	40	2020-01-31
b	3	33	2020-01-31
a	4	27	2020-01-31

Income for each menu-order

orderID	total_price	on_date
1	164.5	2020-01-31
2	85.5	2020-01-31
3	153	2020-01-30
4	120	2020-01-31

Total income for each day

on_date	total_income
2020-01-30	153
2020-01-31	370

Order cost for each order from shop

orderID	order_cost	on_date
1	75.5	2020-01-31
2	10.5	2020-01-30
3	60	2020-01-31

Total spent on each day for shop orders

on_date	total_spent
2020-01-30	70.5
2020-01-31	136

## اطلاعات مربوط به استفاده از ابزارهای پایگاه داده:

برای این پروژه در قسمت طراحی و پیاده سازی دیتابیس رابطه ای از MySQL استفاده شده.

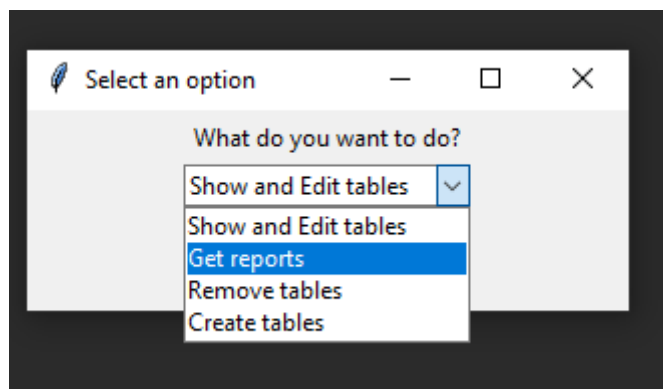
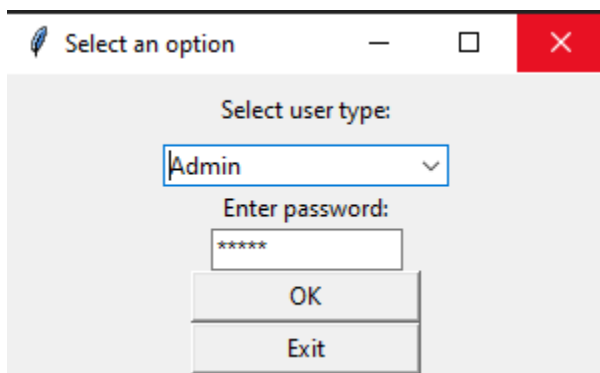
برای برقراری ارتباط با دیتابیس از پایتون استفاده کرده ام که از طریق کتابخانهی `mysql.connector` به دیتابیس وصل می شود و پس از وارد کردن اطلاعات دیتابیس و برقراری ارتباط با آن، اجرای همه ی دستورات دیتابیس از طریق `CURSOR` ای که به ما می دهد قابل اجرا است.

از PyCharm هم به عنوان IDE برای پایتون و مشاهده ی دیتابیس و مشاهده و کنترل رکوردهای هر جدول استفاده کرده ام.

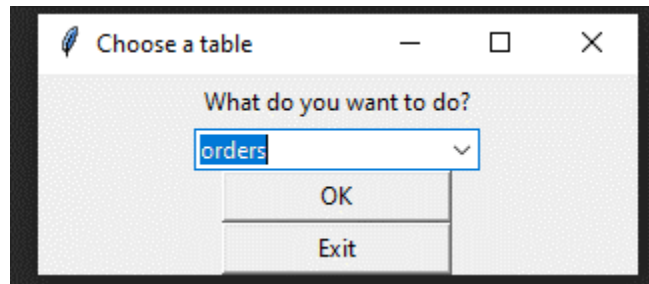
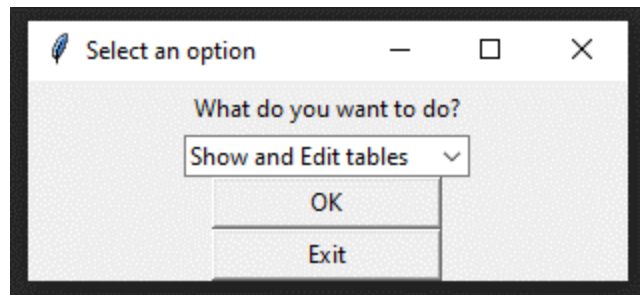
از کتابخانه ی `tkinter` در پایتون برای پیاده سازی رابط کاربری کارا به صورت گرافیکی استفاده کرده ام.

از کتابخانه ی `tabulate` برای پرینت کردن زیباتر نتایج `query` های دیتابیس در ترمینال پایتون استفاده کرده ام.

تمامی عملیات های نمایش اطلاعات جداول، آپدیت/حذف/افزودن رکود از طریق رابط کاربری گرافیکی که نوشته ام قابل اجرا است که چالش های بسیار زیادی را در طول مسیر ایجاد کردند که توضیح آنها در حوصله ی این گزارش نیست ولی نهایتاً برطرف شدند و تصاویری از واسط کاربری گرافیکی که نوشته ام در ادامه قابل مشاهده است:







9631019 - Arash Hajisafi

## orders

	orderID	orderDate	customerID	AID	courierID
0		2020-01-31 17:46:09	2	None	None
1		2020-01-31 17:25:05	3	None	None
2		2020-01-31 17:25:15	1	8	1
3		2020-01-30 17:26:22	2	None	None
4		2020-01-31 20:38:09	3	573	None
7		2020-01-31 20:38:09	46	None	3
30		2020-01-30 17:26:22	None	None	None

### گزارش پرس وجوها (Queryها)

تعداد ۱۰ عدد کوئری نوشته شده که ۸ تای آنها پیچیدگی بسیار قابل توجهی دارند و نتیجه هرکدام و توضیح مربوط به اینکه هرکدام برای چه منظوری هستند در فایل report.txt قرار گرفته و همینطور در برنامه نیز کامنت شده است.