

دانشگاه صنعتي امیرکبیر  
(پلی تکنیک تهران)

دانشكده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه پایانی دیتابیس

نگارش

آرش حاجی صفی - 9631019

استاد درس

دکتر ممتازی

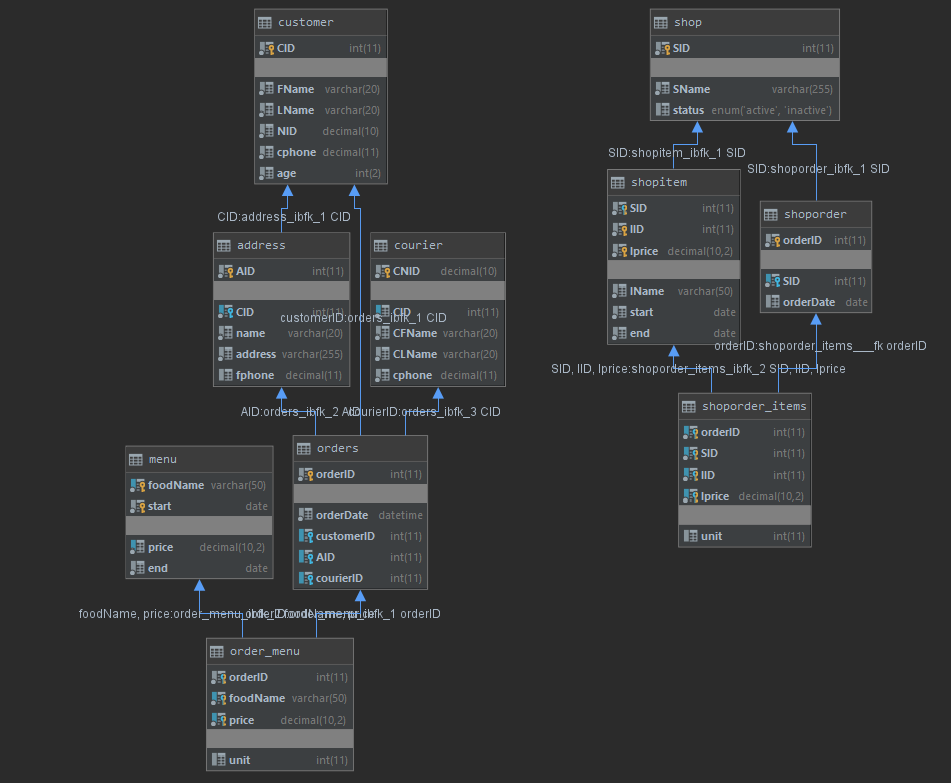
بهمن 1398

طراحی پایگاه داده و حل چالش‌ها :

در مورد طراحی با روش ER در فایل PDF مربوط به مطلوبات 1 تا 4 پروژه توضیح داده شده.

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 داده ام. 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 10 تا هستند که به صورت زیر قرار گرفته اند:



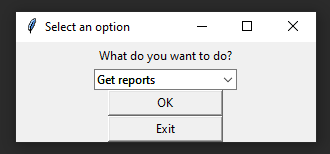
1. جدول customer اطلاعات مشتری را در برمی‌گیرد.
2. یکی از چالش‌ها multivalued بودن آدرس بود که در نتیجه جدول جدایی را برای آدرس گرفته ام که آیدی جدول مشتری در آنها قرار گرفته و هر آدرس متعلق به یک مشتری می‌شود و هر مشتری چندین آدرس می‌تواند داشته باشد.
3. جدول courier اطلاعات پیک را در بر می‌گیرد، هر پیک هم به وسیله‌ی شماره ملی و هم آیدی می‌تواند به طور یونیک مشخص شود.
4. هر رکورد جدول منو شامل اسم غذا، قیمت آن، تاریخ شروع و پایان است. یکی دیگر از چالش ها قابل تغییر بودن اسم و قیمت آیتم های 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 می‌کنند. 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 تاریخ های 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.
5. 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 بیرون‌بر است)، و با courierID به یک پیک مرتبط شود. این 3 فیلد الزامی نیستند و می‌توانند null باشند.
6. جدول order\_menu ارتباط بین سفارش با منو را برقرار می‌کند که یکی دیگر از چالش های 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.
7. جدول shop اطلاعات هر فروشگاه و وضعیت آنرا نگه می‌دارد.
8. جدول shoporder سفارش های رستوران از فروشگاه‌ها و تاریخ انجام سفارش را مشخص می‌کند. منطق این قسمت نزدیک به orders خود رستوران است.
9. جدول shopitem، آیتم‌های هر فروشگاه را در بر دارد. یکی دیگر از چالش ها 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 قیمت ها 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.
10. 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.

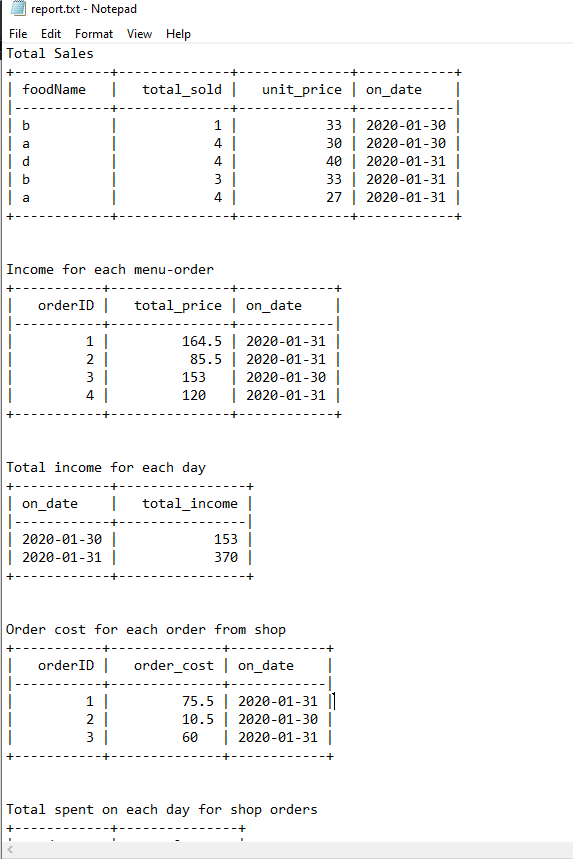
به ازای هر جدول یک جدول هم برای log به وجود آمده که دارای 3 فیلد (نام جدول، نوع عملیات، تاریخ عملیات) به علاوه‌ی تمامی فیلدهای جدول اصلی است که با trigger تمامی عملیات‌های insert/update/delete به همراه رکوردی که تغییر کرده در آن قرار می‌گیرد. پس مجموعاً 10 تا جدول اصلی ایجاد شده است به علاوه‌ی 10تا جدول برای log.

برای بررسی صحت شماره موبایل و تلفن ثابت در جدول‌های address و customer و courier هرکدام یک trigger نوشته شده که برای بررسی شماره موبایل چک می‌شود که حتماً 10 رقم باشد (بدون صفر اول) و به فرم 9xxxxxxxxx باشد و برای تلفن ثابت هم بررسی می‌شود که 10 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 refreshLogs() ایجاد شده که هیچ پارامتر ورودی ندارد و در تمامی جداول log، اگر از تاریخ ایجاد رکوردی بیشتر از 3 روز (نسبت به تاریخ فعلی) گذشته باشد، آن رکوردها را پاک می‌کند.

تمامی queryهای گزارش‌گیری 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 می‌شود:





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 می‌شود و پس از وارد کردن اطلاعات دیتابیس و برقراری ارتباط با آن، اجرای همه‌ی دستورات دیتابیس از طریق cursorای که به ما می‌دهد قابل اجرا است.

از PyCharm هم به عنوان IDE برای پایتون و مشاهده‌ی دیتابیس و مشاهده و کنترل رکوردهای هر جدول استفاده کرده‌ام.

از کتابخانه‌ی tkinter در پایتون برای پیاده‌سازی رابط کاربری کارا به صورت گرافیکی استفاده کرده‌ام.

از کتابخانه‌ی tabulate برای پرینت کردن زیباتر نتایج queryهای دیتابیس در ترمینال پایتون استفاده کرده ام.

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

