



MizDooni

مقدمه

در این پروژه می‌بایست تمام داده‌های مورد نیاز برنامه را که در فازهای قبلی روی حافظه اصلی^۱ ذخیره و استفاده می‌شدند را روی پایگاه داده که حافظه‌ای ماندگار است، نگهداری کنید. ابتدا شما باید شمای پایگاه داده خود را طراحی کنید و پس از خواندن داده‌ها از آدرس <http://91.107.137.117:55> آن را روی پایگاه داده نوشته و هنگام استفاده از پایگاه داده بازیابی کنید. قسمت front-end پروژه در این فاز تغییری نمی‌کند ولی لازم است داده خوانده شده از پایگاه داده را نمایش دهد.

پایگاه داده رابطه‌ای^۲ MySQL برای این پروژه در نظر گرفته شده است. در این نوع پایگاه داده‌ها، داده در ساختار جدول‌هایی با تعداد صفت^۳‌های ثابت با نوع مشخص شده، ذخیره می‌شود. برای طراحی پایگاه داده و ساخت جدول‌ها لازم است از آموخته‌های اولیه خود از درس پایگاه داده مانند کشیدن نمودار Entity-Relationship استفاده کنید. شباهت زیادی مابین صفت موجودیت^۴‌های پایگاه داده و صفت اشیاء دامنه برنامه در طراحی شیء‌گرا^۵ وجود دارد که می‌توانید برای تعریف موجودیت‌ها از آن استفاده کنید. برای این تبدیل در این پروژه الزامیست از ORM‌ها که در درس با آن آشنا شدید، استفاده کنید. دقت کنید که همه داده‌های موجود در برنامه حتی اگر در این فاز از آن‌ها استفاده‌ای نمی‌شوند، باید ذخیره شود؛ زیرا ممکن است در آینده به آن نیاز داشته باشید. در ادامه موجودیت‌های اصلی برنامه برای درک بهتر توضیح داده می‌شوند. اما دقت کنید که نیاز است جدول‌های دیگری از جمله ارتباط^۶‌های نمودار بین موجودیت‌های اصلی را نیز تعریف کنید.

¹ Main memory

² Relational database

³ attribute

⁴ entity

⁵ Object-oriented design

⁶ relationship

نگهداری اطلاعات در پایگاه داده

کاربران

همانطور که در فازهای قبل پروژه پیاده‌سازی کردید، برای هر کاربر سامانه صفت‌های نام کاربری، رمزعبور، ایمیل، آدرس و نقش تعریف می‌شود که باید در جدول کاربران ذخیره شوند. مانند فاز قبل داده را از api مربوطه از آدرس سرویس خارجی خوانده و با دستور مناسب داخل پایگاه داده می‌ریزید. هر دو صفت نام کاربری و ایمیل برای کاربران یکتا است؛ پس یکی از آن دو را به عنوان کلید اصلی⁷ جدول تعریف کنید. کاربران می‌توانند دو نقش مدیر و مشتری داشته باشند که می‌بایست با سلسله مراتب ISA مخصوص نمودار ER از یکدیگر جداگانه تعریف شوند. نمونه‌ای از پیاده‌سازی ارث‌بری⁸ در ORM نیز در اسلایدهای درس بخش JPA موجود است.

رستوران

در جدول رستوران باید صفت‌های نام کاربری مدیر، نوع، ساعت شروع و پایان کار، توضیحات، آدرس و لینک عکس ذخیره شود. این اطلاعات را از سرویس خارجی دریافت کرده و داخل پایگاه داده می‌نویسید. نام رستوران یکتا است که به عنوان کلید اصلی این رابطه تعریف می‌شود.

باید ارتباطی بین رکورد رستوران و رکورد مدیر آن ایجاد شود. این ارتباط در زبان پرس‌وجوی SQL با کلید خارجی⁹ تعریف می‌شود. کلید خارجی در واقع همان کلید اصلی جدول دیگری است و نوع صفت کلید خارجی و کلید اصلی متناظر آن باید یکسان باشد. این ارتباط را به گونه‌ای طراحی کنید که امکان اینکه یک مدیر، در چند رستوران مدیریت کند نیز وجود داشته باشد. توجه کنید که هر رستوران دقیقاً یک مدیر دارد. باید محدودیتی ایجاد کنید که هنگام اضافه شدن یک رستوران، مدیر آن قبلاً به پایگاه داده اضافه شده باشد.

همچنین بهتر است آدرس را به جای یک صفت مرکب، یک موجودیت جداگانه تعریف کنید و یک ارتباط بین این دو موجودیت ایجاد کنید.

میز

در موجودیت میز اطلاعاتی شامل شناسه میز، نام رستوران، نام کاربری مدیر رستوران و تعداد صندلی‌ها ذخیره می‌شود. شناسه میز را به صورت auto increment به وسیله انوتیشن مناسب تولید کنید و از مقدار tableNumber که از سرویس خارجی می‌گیرید صرفه نظر کنید. نام رستوران کلید خارجی به رابطه رستوران و نام کاربری مدیر کلید خارجی به رابطه مدیر باشند.

⁷ Primary key

⁸ inheritance

⁹ Foreign key

بازخورد

موجودیت بازخورد شامل اطلاعات نام کاربری مشتری، نام رستوران، نمره کیفیت غذا، نمره سرویس دهی، نمره محیط رستوران، نمره مجموع، کامنت و زمان ثبت می باشد که در یک جدول ذخیره می شود. این موجودیت یک شناسه یکتا نیز لازم دارد که به صورت auto increment برای آن تولید کنید. نام کاربری مشتری و نام رستوران می بایست کلید خارجی قرار داده شوند.

رزرو

اطلاعات رزرو شامل نام کاربری مشتری، نام رستوران، شماره میز و زمان رزرو است. یک صفت شناسه یکتا نیز به آن اضافه کنید که به صورت خودکار تولید می شود. با قرار دادن محدودیت کلید خارجی مطمئن شوید که چنین نام کاربری و چنین رستوران با چنین شماره میزی در جدول های مربوطه وجود دارند.

چارچوب Hibernate

در این پروژه ملزم به استفاده از چارچوب Hibernate هستید. همانطور که در درس با Java Persistence API آشنا شده اید، یک استاندارد برای تبدیل جدول ها به اشیاء دامنه و برعکس می باشد. در واقع Hibernate معروف ترین شرکت تولید کننده JPA است که تبدیل به معروف ترین ORM موجود در زبان جاوا شده است.

استفاده از چارچوب های ORM از پیچیدگی های مربوط به نگاشت داده های موجود در پایگاه داده به اشیاء دامنه و پیاده سازی پرس و جوهای مختلف می کاهد. گزینه جایگزین ایجاد اتصال توسط درایور JDBC با پایگاه داده است که در درس با آن آشنا شده اید. در این روش باید پرس و جوهای خود را به زبان SQL و در قالب statement داخل سورس برنامه بنویسیم. همچنین پس از دریافت نتیجه پرس و جو در فرمت ResultSet باید آن را دوباره به شیء دامنه برنامه خود تبدیل کنیم. این فرآیند برای تمام موجودیت های پایگاه داده تکرار می شود که باعث نوشته شدن حجم زیادی کد خواهد شد. اما وقتی که از ORM ها استفاده می شود خودشان پرس و جوهای SQL ساخت جدول ها و بازیابی اطلاعات را تولید خواهند کرد.

برای آشنایی با معماری JPA و انوتیشن های مختلف آن برای ایجاد رابطه های مختلف بین موجودیت ها و همچنین ایجاد یک EntityManager به اسلایدهای درس و منابع موجود در اینترنت رجوع کنید. نحوه انجام عملیات های مختلف در JPA نیز توضیح داده شده است که در پیاده سازی منطق برنامه به آن نیاز خواهید داشت. برای ایجاد تغییر در پایگاه داده لازم است یک تراکنش¹⁰ آغاز کرده و در انتها آن را commit کنید.

¹⁰ transaction

نکات تکمیلی و راهنمایی

- در صورت عدم استفاده از Hibernate بخش اعظمی از نمره شما کسر خواهد شد.
- فرآیندهای جستجو باید در سطح پایگاه داده انجام گیرد. زیرا اجرای پرس و جوها توسط DBMS بهینه سازی می شود. یعنی مثلا برای جستجو نام یک رستوران نباید همه رستوران ها را خوانده و سپس در حافظه اصلی یک نام خاص را پیدا کنید. یا نباید همه میزها را وارد حافظه اصلی کرده و سپس میزهای یک رستوران خاص را فیلتر کنید. به جستجو در منطق برنامه نمره ای تعلق نمی گیرد.
- در صورتی که MySQL را روی سیستم خود نصب ندارید، ابتدا آن را نصب کنید. لازم است هنگام اجرای برنامه MySQL بالا باشد تا اتصال به پورت مخصوص آن صورت پذیرد.
- دقت کنید تعریف کلید اصلی و کلید خارجی مناسب برای جدول هایتان ضروری است.
- برای ذخیره سازی اطلاعات مختلف انواع داده¹¹ مناسب در شمای رابطه استفاده کنید.
- برای مطالعه درباره مقادارهای اتوماتیک تولید شده در MySQL به این لینک مراجعه کنید. با استفاده از انوتیشن @GeneratedValue در JPA می توانید برای یک صفت مقدار تولید کنید.
- هنگام خواندن اطلاعات مختلف در شروع برنامه، در صورت وجود موارد جدید آن را به پایگاه داده اضافه کنید و یا در صورت آپدیت شدن اطلاعات، تغییرات لازم را در پایگاه داده اعمال کنید. در غیر این صورت، تغییری در پایگاه داده ایجاد نکنید.
- هنگام استفاده از EntityManager نیازی به مدیریت اتصالها (connection pooling) ندارید.

Software Engineering Best Practices

درباره پوشه بندی و نحوه مدیریت فایل ها هنگام اتصال اپلیکیشن به پایگاه داده تحقیق کنید. این لینک می تواند منبع مناسبی باشد. لازم است قبل از شروع به نوشتن برنامه این کار را انجام دهید زیرا ابتدا ساختار مناسبی برای برنامه خود پیدا کرده باشید. هر کدام از کلاس هایی که به یک جدول نگاشت شده اند، در پوشه entity قرار می گیرند. همچنین برای هر جدول یک کلاس dao وجود دارد مخصوص دسترسی به داده های آن جدول است. این کلاس ها باید از JpaRepository ارث بری کنند. متدهای کلاس dao توسط لایه سرویس فراخوانی می شوند.

Git Commit

همان طور که در پروژه اول توضیح داده شد، کامیت ها اهمیت زیادی در توسعه پروژه های نرم افزاری دارند. در این پروژه نیز باید مواردی که در پروژه اول گفته شدند، رعایت شوند. رعایت این قسمت، بخشی از نمره شما را در این پروژه تعیین می کند.

¹¹ Data type

نکات پایانی

- کافی است که یکی از اعضای گروه Hash مربوط به آخرین کامیت پروژه سمت کاربر و سمت سرور را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت مورد ارزیابی قرار می‌گیرد.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاست درس، نمره کسر خواهد شد.
- حتماً کاربر IE-S03 را به مخزن خود اضافه کرده باشید.
- در صورت داشتن سوال درباره پروژه به آدرس m.hadi.hojjat@gmail.com ایمیل بزنید.

موفق و پیروز باشید.