



به نام خدا

پروژه سوم اصول طراحی پایگاه داده

دکتر حمیدرضا شهریاری

نیمسال دوم ۱۴۰۲-۱۴۰۳



مقدمه

در این پروژه قرار است به پروژه پایگاه داده سامانه رزرو غذای دانشگاه (سماد) چندین ویژگی اضافه کنیم. هدف از این پروژه آشنایی با مفاهیم پیشرفته SQL و پایگاه داده است.

تعریف پروژه

در این پروژه با استفاده از پایگاه داده PostgreSQL می‌خواهیم سیستم رزرو غذای دانشگاه را پیاده‌سازی کنیم. شما مجاز هستید از هر زبان برنامه نویسی استفاده کنید ولی دقت کنید که کوئری‌هایی که پیاده‌سازی می‌کنید حتما باید به صورت Raw باشند و نباید از ORM و ابزارهای کمکی دیگر استفاده کنید.

شرح پیاده‌سازی

سیستم رزرو غذای دانشگاه بدین صورت است که از چند جدول تشکیل شده است که هر کدام ویژگی‌ها و مشخصاتی دارند که در **پروژه دوم** درس به آن پرداختیم. حال می‌خواهیم با استفاده از ویژگی‌های پیشرفته زبان SQL چندین ویژگی جدید به پروژه اضافه کنیم.

پرونده‌ها (Procedures):

[پرونده‌ها](#) مانند توابع در زبان‌های برنامه نویسی هستند و در این بخش قصد داریم تا برای چند تا از SQL Statement های مهم پروژه‌مان پرونده بنویسیم تا بتوانیم در وضعیت‌های گوناگون از آن استفاده کنیم.

- ۱) پرونده‌ای بنویسید که زوج نوع و تعداد را برای غذاهای رزرو شده امروز نشان دهد.
- ۲) پرونده‌ای بنویسید که لیست افرادی که امروز غذا رزرو کرده‌اند را نشان دهد.
- ۳) پرونده‌ای بنویسید که لیستی از ۱۰ تراکنش اخیر سیستم را نشان دهد.
- ۴) پرونده‌ای بنویسید که زوج نوع و تعداد را برای غذاهای باقی مانده امروز نمایش دهد. (غذاهایی که تعداد رزروهایشان کمتر از inventory آنها باشد)
- ۵) پرونده‌ای بنویسید که گردش مالی روزانه سلف دانشگاه را نشان دهد. (مجموع قیمت غذاهای رزرو شده برای یک روز)



تریگرها (Triggers):

در برخی موارد نیاز داریم تا با رخ دادن یک تغییر در دیتابیس، عملیاتی به صورت خودکار انجام شود تا عملیاتی را قبل یا بعد از انجام آن تغییر بر روی داده‌ها انجام دهد. بدین منظور نیاز داریم تا با تعریف تریگرهای مناسب برای جدول‌های مربوط به آنها از روی دادن موارد زیر جلوگیری کنیم:

- (۱) اضافه شدن دانشجو با کد ملی یا دانشجویی یکسان
- (۲) کمتر از ۵۰۰ بودن موجودی اولیه هر غذا
- (۳) سرو شدن بالای ۲ نوع غذا در یک زمان
- (۴) سفارش بیش از یک غذا توسط یک دانشجو در یک زمان

ویوها (Views):

- در این قسمت قصد داریم تعدادی view به شرح زیر به پایگاه داده اضافه کنیم:
- یک view با نام student_food تعریف کنید که نام همه غذاهای رزرو شده توسط دانشجو را در کنار نام او نشان دهد.
 - یک view با نام student_transaction بنویسید که در کنار نام دانشجو و شماره دانشجویی‌اش، اطلاعات رزروی که تغییر داده است (شامل نام غذای رزرو اولیه و غذای رزرو جدید) ذکر شود.

بهبود عملکرد و سرعت پردازش کوئری‌ها (امتیازی):

می‌دانیم که اکثر کوئری‌هایی که برای یافتن دانشجویان در جدول students اجرا می‌شوند، جست‌وجو را بر اساس ستون studentID انجام می‌دهند. چه راهکاری برای بهبود این کوئری‌ها و افزایش سرعت پردازش آنها پیشنهاد می‌کنید؟

راهکار خود را به شکل قطعه کدی به زبان SQL ارائه کرده، و با اجرای یک کوئری نمونه که دانشجویی را با شماره دانشجویی او جست‌وجو می‌کند، زمان اجرای کوئری را در حالت اولیه و حالت بهبود یافته اندازه‌گیری و مقایسه کنید. نتایج حاصل را در گزارش خود ثبت کنید.



راهنمایی ۱: در رابطه با مفهوم Index در پایگاه داده‌های رابطه‌ای اطلاعات کسب کنید.
راهنمایی ۲: ممکن است نتایج حاصل از بهبود در جداولی که ردیف‌های کمی دارند محسوس نباشد.

جلوگیری از ورود داده‌های نامعتبر (امتیازی):

با توجه به اینکه مقدار ستون‌های SRCreservationID و DSTreservationID باید به ازای هر یک از ردیف‌های جدول transactions یکتا باشند، یعنی هیچ دو ردیفی مقدار یکسانی در هر دوی این ستون‌ها نداشته باشند، چطور می‌توانیم از درج داده‌های نامعتبر جلوگیری کنیم؟

منظور از داده‌های نامعتبر، ردیف‌هایی است که یکتا بودن مقادیر این دو ستون را نقض می‌کنند و هدف این است که هر گاه دستور درج داده‌ای داده شود که مقدار هر دو ستون گفته شده آن با یکی از ستون‌هایی که پیش‌تر در جدول هست یکسان است، از ثبت آن جلوگیری شود. راهکار خود را به شکل قطعه کدی به زبان SQL ارائه کنید.

راهنمایی: در رابطه با مفهوم Unique Index در پایگاه داده‌های رابطه‌ای اطلاعات کسب کنید.

پیاده سازی کش (امتیازی):

می‌توانید از Redis به عنوان cache و به منظور بهبود عملکرد و سرعت درخواست‌های پایگاه داده استفاده کنید. به عنوان مثال، با ذخیره‌سازی اطلاعاتی مانند موجودی غذاها و موجودی حساب دانشجویان در این ابزار، می‌توانید از دسترسی‌های مکرر به پایگاه داده PostgreSQL جلوگیری کرده و سرعت دسترسی به داده‌ها را افزایش دهید.

برای این کار، می‌توانید برای هر موجودی غذا و موجودی حساب دانشجویی یک کلید در Redis ایجاد کرده و هر گاه که رزرو جدیدی صورت می‌گیرد یا تغییری در موجودی غذا یا موجودی حساب دانشجویی اعمال می‌شود، مقدار آن را بروزرسانی کنید.

راهنمایی: نکته حائز اهمیت این است که باید همواره سازگاری بین دیتابیس اصلی و داده‌های cache شده حفظ شود. از این رو در رابطه با مفهوم consistency اطلاعات کسب کنید.



نکات پیاده سازی:

- برای این فاز از پروژه سیستم مدیریت پایگاه داده PostgreSQL یا به اختصار Postgres در نظر گرفته شده است.
- برای تمامی موارد گفته شده، استفاده تنها از raw query مجاز می‌باشد و مجاز به استفاده از ORM نمی‌باشید.
- هنگام پیاده سازی عملیات‌ها، باید تمامی حالت‌های خطا را مدیریت کنید و پیغام‌های مناسبی نمایش دهید.
- نیازی به واسط کاربری گرافیکی وجود ندارد.
- در صورت بروز هرگونه ابهام یا مشکل در رابطه با پروژه، با [تدریس‌یاران در گروه پرسش و پاسخ](#) در ارتباط باشید.

مواردی که باید در گزارش کار شرح دهید:

- اسکرین‌شات از اجرای برنامه و بخش‌های پیاده‌سازی شده
- توضیح نحوه پیاده‌سازی پروسه‌ها و تریگرها و نتیجه اجرای آن‌ها
- خروجی برنامه در هنگام اجرای دستوراتی که باعث فعال‌سازی تریگرها می‌شود
- گزارش مربوط به نحوه بهبود پردازش کوئری‌ها و مقایسه عملکرد حاصل (امتیازی)
- گزارش مربوط به نحوه جلوگیری از ورود داده‌های نامعتبر (امتیازی)
- گزارش مربوط به پیاده‌سازی ویوها و نمایش آن‌ها (امتیازی)
- گزارش مربوط به پیاده‌سازی cache و نحوه حفظ consistency (امتیازی)



نکات قابل توجه:

هر یک از بخش‌های امتیازی این پروژه، می‌توانند جبران کننده نمراتی که در پروژه‌های گذشته از دست دادید باشند. توجه کنید امتیازی مربوط به پروژه تنها بخش پروژه را جبران می‌کند و تأثیری در نمره سایر بخش‌های درس ندارد.

توجه کنید استفاده از بودجه تأخیر در این پروژه مجاز نمی‌باشد بنابراین، تحویل با تأخیر به هیچ وجه پذیرفته نیست.

۱. تمامی فایل‌های خود را در قالب pdf به صورت یک فایل آرشیو zip که با قالب زیر نام‌گذاری شده است؛ اپلود بفرمایید.

DB-Project3-FirstName_LastName-StudentNumber.zip

۲. در این پروژه مشورت با دوستان، همکلاسی‌ها، و تدریس‌یاران درس کاملاً آزاد است اما توجه داشته باشید که در نهایت پروژه شما به صورت انفرادی بررسی خواهد شد و شما باید درک خود را از پروژه تحویل دهید.

۳. این پروژه تحویل به صورت مجازی در google meet یا skype خواهد داشت و عدم حضور در جلسه به معنای از دست دادن تمام نمرات امتیازی و ۷۰ درصد از نمرات اصلی خواهد بود.

۴. در صورت وجود هرگونه ابهام یا سوال می‌توانید از طریق گروه پرسش و پاسخ با تدریس‌یاران درس در ارتباط باشید.

۵. مهلت تحویل پروژه تا ۱۰ تیر ساعت ۲۳:۵۹ خواهد بود و امکان تمدید مهلت تحویل پروژه به هیچ وجه وجود ندارد.