

بسم الله الرحمن الرحيم

جزئیات طراحی سیستم

فاطمه شاه علی ۹۶۱۰۰۴۰۳ و نیوشا سیدی ۹۶۱۰۸۶۴۵

- محدودیت‌های دامنه‌ای را در تعریف صفات هر جدول، مشخص کرده ایم.
- محدودیت‌های صفتی:
 - در جدول‌های Bookstore و Book و Comment و User، مقادیر امتیاز باید بین ۰ و ۱۰ باشد.
 - در جدول Transaction، صفت Type (نوع تراکنش) تنها می‌تواند یکی از مقادیر IncreaseIncome، WithdrawAmanat، IncreaseAmanat و IncreaseExpenditures را داشته باشد. همچنین، سقف مقدار تراکنش (Amount) به میزان ۵۰۰۰ دلار است.
 - در جدول Borrow، صفت Status (وضعیت قرض) تنها می‌تواند مقادیر Borrowed، Cancelled، Returned، InDelay و GetGuaranteePrice را داشته باشد.
 - در جدول UserCommentForUser (نظر به کاربر)، صفت ReceiverType (نوع کاربر نظرگیرنده) می‌تواند یکی از مقادیر Borrower، Lender، Seller، Purchaser باشد.

● Triggerها:

- سطرهای جدول Transaction قابل حذف کردن نیستند.
 - سطرهای Transaction قابل تغییر نیست.
 - تاریخ آخرین تسویه حساب با کتاب فروشی هیچگاه قبل از تاریخ قبل آخرین تسویه حساب با کتاب فروشی نمی شود.
 - موفقیت یک خرید نمی تواند از ۱ به ۰ تغییر کند.
 - تأییدیه قرض، نمی تواند از ۱ به ۰ تغییر کند.
 - در جداول User و Bookstore و Manager و Employee، طول رمز عبور (Password) نباید کمتر از ۸ کاراکتر باشد.
 - همزمان با ساخت اکانت برای یک کاربر، یک کیف پول نیز برای آن کاربر ساخته می شود.
 - همزمان با ساخت اکانت برای یک کتابفروشی، حساب بانکی مربوط به آن کتابفروشی در سیستم نیز ساخته می شود.
 - بعد از دریافت نظرات مربوط به کاربرها، کتاب ها و کتابفروشی ها، میانگین امتیاز مربوط به هر کدام (که در کاربر، نوع آن یعنی امتیاز از نظر قرض دهنده، قرض گیرنده، فروشنده یا خریدار است) باید محاسبه شود و در جدول مربوطه به روزرسانی شود.
 - بعد از تایید هر قرض، باید هزینه قرض و مبلغ تضمین تحویل کتاب در کیف پول کاربر قرض گیرنده از حساب قابل دسترس به حساب بلوکه شده منتقل شود.
- محدودیت های جامعیتی:

- در راستای ارضای محدودیت های جامعیتی و نیز تحقق اهداف، روشی را انتخاب کردیم که ابتدا تمام جداول را می سازد و هیچ کلید خارجی ای در آن تعریف نشده است. این

کار مربوط به فایل CreateTables.sql است. سپس تمام کلیدهای خارجی را که روابط را مشخص می کنند، در فایل CreateRelations.sql تعریف کرده ایم. همچنین برای حذف کردن جدول ها، ابتدا تمام رابطه ها را حذف می کنیم، سپس خود جدول ها را حذف می کنیم که کد آن در Remove.sql قرار دارد. بنابراین برای اعمال جامعیت ارجاعی در حذف، از روش RESTRICTED استفاده کرده ایم. خوشبختانه MySQL الزامی بر اینکه کلیدهای خارجی حتما Not Null باشد، ندارد و این باعث می شود که دور ایجاد نشود. اگر از ابزاری استفاده می کردیم که این مشکل را داشت، می توانستیم ابتدا داده ها را وارد کنیم، سپس رابطه ها را برقرار کنیم. اگر بخواهیم محدودیت های جامعیتی ارجاعی را رعایت کنیم، می توانیم نگار صفاتی که کلید خارجی هستند، Not Null بگذاریم. اما از لحاظ ماهیت مسئله بعضی اوقات واقعا اطلاعات آن در دسترس نیست و اهمیتی ندارد برای همین می تواند Null باشد. مثلا در مورد کتاب، ما کلید ارجاعی ناشر را داریم. اما اگر ندانیم که ناشر آن کیست هم باز می خواهیم بتوانیم داده های آن کتاب را داشته باشیم. برای همین در عمل در سیستم ما لزوما محدودیت جامعیتی ارجاعی رعایت نشده است و این بر اساس نیاز بوده است.

- در مورد محدودیت های جامعیتی موجودیتی هم که با تعریف کردن کلید اصلی در کدمان، این محدودیت خود به خود برقرار میشود و برای تسهیل کار هم کلیدهایی که به صورت id هستند به طور خودکار پر می شوند.

- نرمال سازی جدول ها:

- با توجه به اینکه در تمام مراحل سعی کردیم که جدول هایمان به حد کافی ساده باشند با اینکه مفهوم نرمال سازی را بلد نبودیم، عملا جدول هایمان در سطح نرمال BCNF هستند و نیازی به تغییرات نداشتند.

- دو باهم روی پروژه کار کردیم.