


<p>پروژه درس اصول و طراحی پایگاه داده دکتر ممتازی ترم پائیز ۱۳۹۹ – دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر زمان تحویل: ۹۹/۱۱/۲۰</p>	
--	---

لطفا پیش از شروع به حل کردن تمرین به نکات زیر توجه فرمایید:

۱. در صورت وجود هرگونه سوال یا ابهام می‌توانید با تدریسار از طریق ایمیل [alirezasalehy@aut.ac.ir](mailto:alirezasalehy@aut.ac.ir) در ارتباط باشید.
۲. مهلت تحویل پروژه تا تاریخ ۹۹/۱۱/۲۰ می‌باشد.
۳. همه کد های SQL خود را در یک فایل sql قرار دهید.
۴. فایل sql را به همراه فایل های برنامه رابط کاربری به همراه تمامی کتابخانه های استفاده شده (به صورتی که در روی دستگاه های دیگر قابل اجرا باشد)، در قالب یک فایل zip با فرمت زیر در سایت بارگذاری نمایید. "StudentNumber>.zip"

در این پروژه می بایست پایگاه داده ای را که برای کتابخانه زنجیره ای در تمرین 3 طراحی کردید را پیاده سازی کنید. در صورت نیاز به بازخوانی تعریف پروژه می توانید به تمرین 3 مراجعه کنید. توجه داشته باشید که در این پروژه برای سادگی فرض می شود که تنها یک شعبه و تنها یک انبار داشته باشیم. در نتیجه نیازی به جدول برای انبار و شعبه نیست.

هدف از این صورت پروژه، پیاده سازی مدل پایگاه داده، کار با SQL به صورت مستقیم و پیاده سازی روابط و مدیریت داده ها و کار با مدل پیاده سازی شده از یک رابط کاربری جداگانه می باشد.

توجه داشته باشید که ساختار هایی که برای موجودیت ها گفته می شود ممکن است کامل نباشد و شما باید بر اساس صورت پروژه که در تمرین سه بود و صلاح دید خود ویژگی هایی را به آن بیافزایید. ولی در معماری خود اصول نرمال سازی شماتیک پایگاه داده برای پیشگیری از افزونگی و... را به کار ببرید.

در این پروژه باید برای همگی توانمندی های خواسته شده یک تابعی مجزا با ورودی های مناسب تعریف کنید و که این تابع ها بدست رابط کاربری فراخوانی می شوند.

توجه داشته باشید که می توانید نرم افزار های DBMS دلخواه مانند؛ MariaDB یا MySQL را برای پیاده سازی پایگاه داده و هر یک از زبان های برنامه نویسی دلخواه برای پیاده سازی رابط کاربری، به کار ببرید.

نکات کلی:

- همه مورد های خواسته شده باید کاملاً در SQL پیاده سازی شوند، به این صورت که رابط کاربری تنها خروجی ها گرفته شده از پایگاه داده را سامان و نمایش می دهد.
- برای نگه داری رشته های طولانی می توانید طول 512 را در نظر بگیرید.
- برای نگه داری شماره تلفن و یا شماره های دیگر از ویژگی های عددی استفاده نکنید. زیرا صفر آغازین آن ها از دست می روند.
- مواردی که با "به صورت خودکار" توصیف شده اند، باید با trigger پیاده سازی شوند.
- موارد امتیازی با سبز نمایش داده شده اند.

## حساب کاربری

هر مشتری دارای یک حساب کاربری می باشد، اطلاعات هر فرد به دو بخش اطلاعات سیستمی مانند "نام کاربری و گذرواژه" و اطلاعات شخصی مانند "نام آدرس شماره تلفن و..." دسته بندی می شود.

## اطلاعات سیستمی :

این بخش داده های زیر را در بر دارد:

- نام کاربری:

یکتا می باشد و باید دست کم 6 کاراکتر از میان حروف الفبا و رقم باشد، نام کاربری به هنگام نمایش داده شدن حساس به کوچک و بزرگ بودن حرف می باشد و به هنگام احزار هویت یا ساخت حساب کاربری بزرگ یا کوچک بودن حرف ها مهم می باشد. برای نمونه دو کاربر با نام کاربری farhad و Farhad مجاز نمی باشد. و همچنین اگر کاربری که با نام Farhad ثبت نام کرده باشد به هنگام نمایش به کاربر باید به همین صورت نمایش داده شود.

- گذرواژه :

گذرواژه نیز باید دست کم 8 کاراکتر دربردازننده حروف الفبا و رقم باشد (هر دو)، گذرواژه به هنگام دریافت باید در پایگاه داده به صورت درهم ریخته (Hashed) نگه داری شود. و در هنگام ورود هر فرد به حساب خود باید گذرواژه وارد شده درهم ریخته شود و با آنچه در پایگاه داده ذخیره شده است سنجیده شود، برابر بودن این دو مقدار درهم ریخته نشان دهنده درستی گذرواژه می باشد.

- زمان ساخت حساب:

تاریخ ساخت حساب هر فرد در هنگام ثبت نام باید در پایگاه داده نگه داری شود.

## اطلاعات شخصی:

این بخش داده های زیر را در بر دارد:

- آدرس مکان زندگی

- نام

- نام خانوادگی

- نوع کاربری (دانشجو، استاد، عمومی)

- و...

### ثبت نام:

مشتری ها برای اینکه بتوانند از خدمات سامانه بهره‌مند شوند نخست باید حساب کاربری بسازند. برای این کار کاربر باید همه داده های اطلاعات شخصی و سیستمی به جز تاریخ ثبت نام و شماره آیدی خود را وارد کنند. عملیات ثبت نام باید در یک تراکنش انجام شود (Transaction) که چنانچه مشکلی پیش آمد و مثلاً ورودی های کاربر ویژگی ها گفته شده برای اطلاعات حساب کاربری را نداشت، برای نمونه؛ نام کاربری تکراری باشد ("Farhad" و "farhad") یا اینکه گذرواژه هم رقم و هم حرف را در بر نداشته باشد، اگر عملیاتی بر روی جدول ها انجام شده بود بازگردانی شود.

توجه شود که همان گونه که بالا تر گفته شد همه پیاده سازی ها و حتی پیام های خطا نیز باید در SQL پیاده سازی شود و رابط کاربری تنها خروجی را در صورت موفقیت آمیز بودن یا روبه رو شدن با خطا به کاربر نشان می دهد.

پیام های خروجی باید متناسب با خطا رخ داده باشند.

### ورود به حساب:

کاربران برای ورود باید نام کاربری و گذرواژه خود را در رابط کاربری وارد کنند. در SQL بررسی می شود که آیا در هم ریخته گذرواژه وارد شده برابر با مقدار در هم ریخته نگه داری شده برای حساب کاربر می باشد یا نه. و اگر برابر نبودند پیام مناسب را بر می گردانند. توجه داشته باشید که احراز هویت (وارد کردن نام کاربری و گذرواژه) تنها باید یک بار و در ورود کاربر به حساب خود انجام شود. و انجام عملیات های دیگر نام و گذرواژه کاربر نباید دوباره به پایگاه داده فرستاده شود.

برای این کار می توانید شناسه ای را به عنوان خروجی عملیات ورود باز گردانید که این تگ می تواند در هم ریخته "نام کاربری + گذرواژه + برچسب زمانی درخواست ورود" باشد که با عملیات ورود کاربر به حساب کاربری به وی اختصاص داده می شود و با عملیات خروج از حساب شناسه منقضی خواهد شد.

در عملیات های بعدی تگ گرفته شده در عملیات ورود، به همراه دیگر مقدار های لازم برای آن عملیات به پایگاه داده فرستاده میشود. و به این صورت مشخص می شود کدام کاربر عملیات مورد نظر را انجام میدهد.

یا به جای استفاده از تگ میتوان عملیات های درخواست شده از پایگاه داده مانند افزایش موجودی را برای آخرین فردی که به سامانه وارد شده است انجام دهید. برای اینکار می توانید از یک جدول استفاده کنید زمان ورود هر

کاربر به حساب کاربری رو خود را ثبت میکند و برای انجام عملیات افزایش موجودی، عملیات را برای آخرین کاربر وارد شده به حساب انجام می دهد. (بدیهی است این روش نسبت به استفاده از تگ محدودیت دارد).

### گرفتن اطلاعات حساب:

هر کاربر می بایست بتواند اطلاعات شخصی و سیستمی حساب خود را ببیند.

### عملیات های کاربر

کاربر می تواند عملیات های گرفتن کتاب، پس دادن کتاب، جستجو کتاب، افزایش موجودی را انجام دهد. نوع کاربری کاربران سطح دسترسی آن ها را مشخص می کند.

### جستجو کتاب:

کاربران می توانند با نام کتاب، نام نویسنده، نسخه و زمان چاپ کتاب ها را جستجو کنند. توجه داشته باشید که هر ترکیبی از 4 ویژگی ممکن است در جستجو به کار برود. برای نمونه "نام نویسنده + نام کتاب" یا "نام نویسنده + زمان چاپ". نتیجهی جستجو باید بر اساس نام کتاب مرتب شده باشد.

### گرفتن کتاب:

کاربران بر حسب نوع حساب کاربری خود می توانند بر مجموعه ای از کتاب ها دسترسی داشته باشند به این صورت که کاربر عمومی می تواند همه کتاب ها مگر کتاب های آموزشی دانشگاهی و کتاب های مرجع، دانشجو همه کتاب ها مگر کتاب های مرجع و استاد نیز همه کتاب ها را درخواست دهد. (با اینکه این دسته بندی ممکن است غیر واقعی باشد ولی برای بالاتر بردن کیفیت آموزشی این پروژه در نظر گرفته شده است)

همچنین در گرفتن کتاب موجودی حساب کاربر و تاریخچه وی و موجودی انبار نیز بررسی می شود به این صورت که کاربر با 4 بار دیرکرد در پس دادن کتاب در بازه دوماهه اخیر، به اندازه یک ماه محروم می شود.

هزینه گرفتن هر کتاب نیز به اندازه 5 درصد بهای کتاب می باشد و هزینه گرفتن کتاب از موجود حساب وی کاسته می شود. اگر موجودی حساب کاربر کمتر از این مقدار داشت نمی تواند کتاب را بگیرد.

همان گونه که پیش از این گفته شد همه این موارد به همراه خروجی مناسب در صورت موفقیت یا شکست عملیات باید در SQL پیاده سازی شود.

با موفقیت آمیز بودن درخواست باید یکی از کتاب های انبار کاهش یابد. چه عملیات موفقیت آمیز باشد و چه نباشد باید در تاریخچه ثبت گردد. و نتیجه نیز می تواند موفقیت آمیز بودن، محرومیت یا نبود موجودی مالی یا نبود کتاب باشد باید در تاریخچه ثبت گردد. (برای سادگی فرض کنید نتیجه تنها یکی از این ها خواهد بود. یعنی به ترتیب سه مشکل محرومیت، نبود موجودی مالی یا نبود کتاب بررسی می شود و در صورت مشاهده اولین مشکل همان ثبت می شود. مثلا اگر هم موجودی حساب کافی نبود و هم کتاب نبود عدم موجودی حساب ثبت شود).

همچنین به صورت خودکار برای برای مسئولین کتابخانه پیام رویداد درخواست موفقیت آمیز کتاب با جزئیات آن ثبت می شود. (به صورت متنی، مانند در تاریخ A کاربر B کتاب C را به صورت موفقیت آمیز درخواست داده است).

### تحويل کتاب:

با تحويل کتاب به کتابخانه یک واحد به موجودی انبار افزوده می شود. و این عملیات در تاریخچه ثبت میگردد. همچنین به صورت خودکار پیامی برای توصیف رویداد در صندوق پیام مسئولین ثبت می شود. که در این پیام با تاخیر بودن یا بدون تاخیر بودن این تحويل نیز نمایش داده می شود.

### افزایش موجودی:

کاربر می تواند موجودی خود را افزایش دهد، به بازه ای که به عنوان ورودی به تابع مربوط به این عملیات داده می شود نیز باید دقت شود و مثلا اگر ورودی تابع برابر با -5 باشد پیام خطای مناسبی بازگردانده شود.

### کتاب

کتاب ها دارای ویژگی های گفته شده در صورت پروژه می باشند.

### مسئول کتابخانه

کتاب خانه می تواند یک یا چندین مسئول کتابخانه داشته باشد که همانند کاربران دارایی حساب کاربری می باشند و نوع کاربری برای حساب آن ها نیز مدیر و کتابدار می باشد.

کتابدار و مدیر این توانایی را دارند که کتاب هایی را به انبار بافزاید، همچنین می توانند کاربران کتابخانه را نیز جستجو کنند و تاریخچه و اطلاعات شخصی آن ها را ببینند. مسئولین کتابخانه یک صندوق پیام مشترک دارند که رویداد های درخواست های پذیرفته شده برای گرفتن کتاب و پس دادن کتاب در آن ثبت می شود.

ولی تنها مدیر می تواند حساب کاربری یک کاربر را حذف کند. پس از حذف حساب کاربری یک کاربر باید همه ردیف های مربوط به آن کاربر در جدول های مختلف پاک گردد. (می توانید از کلید خارجی برای این کار بهره ببرید)

#### افزودن کتاب:

مسئول کتابخانه می تواند چند نسخه از یک کتاب را به موجودی انبار بیافزاید. یا در صورت نبود کتاب یک ردیف تازه برای آن در نظر بگیرد.

#### لیست درخواست های گرفتن کتاب پذیرفته شده:

مسئول کتابخانه می بایست بتواند درخواست هایی را که موفقیت آمیز بوده اند ببیند. لیست درخواست ها باید به صورت صفحه بندی شده و بر اساس تاریخ از زودترین به دورترین مرتب شده باشد. و اگر 10 درخواست وجود داشته باشد با درخواست صفحه 1، 5 نتیجه نخست و با درخواست صفحه 2، 5 نتیجه دیگر نمایش داده شود.

#### لیست کتاب هایی که زمان تحویل آن ها گذشته:

مسئول کتابخانه باید بتواند کتاب هایی را که زمان تحویل آن ها گذشته است ببیند. کتاب ها بر اساس بیشترین تاخیر به کمترین تاخیر باید فهرست شوند.

#### تاریخچه افرادی که یک کتاب را گرفته اند:

مسئول باید این توانایی را داشته باشد که تاریخچه گرفتن و پس داده شدن یک کتاب را مشاهده کند. تاریخچه باید به ترتیب زمانی از نزدیک ترین به دور ترین مرتب شده باشد.

#### جستجوی کاربران:

کاربر مسئول کتابخانه باید بتواند کاربران را بر اساس نام کاربری یا نام خانوادگی جستجو کند. در صورت وجود بیش از یک نتیجه (جستجو با نام خانوادگی) باید به ترتیب نام خانوادگی مرتب شده باشند. و نتیجه ها صفحه بندی شده باشند به این صورت که اگر 10 حساب در نتیجه جستجو بود، با درخواست صفحه یک 5 نتیجه نخست و با درخواست صفحه دوم 5 نتیجه دیگر خروجی داده شود.

توجه داشته باشید که تنها کاربران مسئول باید بتوانند دیگر کاربران را جستجو کنند یا اطلاعات آن ها را ببینند و این مورد باید در کد SQL شما بررسی گردد.

### دیدن اطلاعات شخصی و تاریخچه کاربر:

کاربر مسئول می تواند اطلاعات شخصی و تاریخچه عملیات های کاربران را ببیند. تاریخچه عملیات ها باید بر اساس تاریخ از نزدیک ترین به دور ترین مرتب شده باشد.

## کار با رابط کاربری

- برای پیاده سازی رابط کاربری می توانید از زبان برنامه نویسی یا اسکریپت دلخواه خود بهره ببرید.
- تنها کار رابط کاربری تفسیر ورودی ها را کاربر و فراخوانی تابع های SQL با ورودی های مناسب و ساماندهی (در صورت نیاز) و نمایش خروجی این تابع ها می باشد.
- برای رابط کاربری پیاده سازی برنامه که با کنسول ورودی و خروجی بگیرد، بسنده می کند.
- خروجی می باید به صورت جدول باشد که گونه ای که نام ستون ها (ویژگی ها) و مقدار آن ها به طور مناسبی نمایان باشد. (چه در GUI و چه در CLI)

## نکات پایانی

- افرادی که رابط کاربری را پیاده سازی نمی کنند و تنها بخش SQL را انجام می دهند. 80 درصد نمره را خواهند گرفت.
- افرادی که تنها رابط کاربری را پیاده سازی کنند، نمره ای دریافت نخواهند کرد.
- منطق سامانه نباید در رابط کاربری پیاده سازی شود.