

## به نام خدا

### توضیحات مربوط به اسکیمای Education

#### 1. Triggers

- CreateLibraryMembership

پس از اضافه شدن هر دانشجوی جدید به سیستم، برای او یک حساب کاربری فعال در کتابخانه ایجاد میکند.

- UpdateLibraryAccess

اگر وضعیت دانشجو به "Graduated"، "Withdrawn" یا "Expelled" تغییر کند، حساب کتابخانه او را به طور خودکار غیرفعال میکند.

- CheckNationalID

هنگام ثبت یا ویرایش اطلاعات دانشجو، صحت کدملی را با استفاده از تابع IsValidNationalID بررسی میکند. اگر کدملی نامعتبر باشد، از ثبت یا ویرایش اطلاعات جلوگیری میکند.

- CheckExpellee

بعد از ثبت نهایی تمام نمرات یک ترم، برای دانشجو فعال میشود. اگر معدل دانشجو در دو ترم متوالی کمتر از ۱۲ شده باشد، وضعیت تحصیلی او را به طور خودکار به "Expelled" تغییر میدهد و این رویداد را در جدول Change\_Log ثبت میکند.

- CheckGraduationAfterGrade

بعد از ثبت هر نمره، بررسی میکند که آیا دانشجو تمام واحدهای لازم برای فارغ التحصیلی را گذرانده است یا نه. اگر شرایط فارغ التحصیلی را داشت و هیچ بدهی‌ای به کتابخانه نداشت و در حال حاضر هیچ کتابی از کتابخانه را امانت نگرفته باشد، وضعیت او را به "Graduated" تغییر میدهد؛ در غیر این صورت، مانع فارغ التحصیلی او میشود.

- LogEnrollmentInsert

هر بار که یک دانشجو در درسی ثبت نام میکند، این رویداد را به همراه جزئیات (شماره دانشجو و کد درس) در جدول Change\_Log ثبت میکند.

- PreventEnrollment

این تریگر از ثبت نام دانشجو در درسی که ظرفیت آن تکمیل شده جلوگیری میکند. اگر ثبت‌نامی بیش از ظرفیت کلاس انجام شود، آن را حذف کرده و یک پیام خطا نمایش میدهد.

- LogEnrollmentDelete

هر بار که یک ثبت نام از سیستم حذف میشود، این عمل را به همراه جزئیات کامل در جدول Change\_Log ثبت میکند.

- AssignAdvisorOnStudentInsert

به محض اضافه شدن یک دانشجوی جدید، به طور خودکار یک استاد راهنما از همان دپارتمان دانشجو به صورت تصادفی برای او انتخاب و ثبت میکند.

## 2. Stored Procedures

- Report\_110

این پراسیجر یک گزارش کامل از اطلاعات دروس ارائه شده تولید میکند. کاربر میتواند با فیلترهایی مانند سال، ترم، نام درس، نام استاد یا نام دانشکده، جستجویش را دقیق تر کند. خروجی شامل جزئیاتی مثل مشخصات کلاس (ساختمان و اتاق)، لیست دستیاران آموزشی (TA)، و اطلاعات امتحان (نوع، تاریخ و محل) است. این پراسیجر یک رشته متنی (@sql) تعریف میکند که شامل کوئری اصلی SELECT است. کوئری با 1=1 WHERE شروع میشود؛ که اجازه میدهد بدون نگرانی از اینکه کدام شرط اول است، شروط بعدی با AND اضافه شوند. سپس، پراسیجر تک پارامترهای ورودی (مثل @course\_title\_, @year\_) را با دستور IF چک میکند. اگر هر پارامتر مقداری داشته باشد (یعنی NULL نباشد)، یک شرط AND متناسب با آن به انتهای رشته کوئری اضافه میشود. در نهایت، پس از اضافه شدن تمام فیلترهای مورد نیاز، دستورات GROUP BY و ORDER BY نیز به رشته اضافه شده و کوئری نهایی با sp\_executesql اجرا میشود.

- StudentTermGrades

این پراسیجر برای نمایش نمرات یک دانشجو طراحی شده و دو حالت کاری دارد:

1. گزارش نمرات یک ترم خاص: اگر علاوه بر ID دانشجو، سال و ترم مشخصی را وارد شود، پراسیجر دو نتیجه برمیگرداند. ابتدا، لیستی از تمام درس‌های آن ترم به همراه نمره و تعداد واحد هر کدام را نمایش میدهد. سپس، معدل (GPA) دانشجو در همان ترم خاص را محاسبه کرده و نشان میدهد.
2. گزارش کلی سوابق تحصیلی: اگر فقط ID دانشجو را وارد شود، پراسیجر خلاصه‌ای از کل سوابق تحصیلی او را نمایش میدهد. خروجی آن به ازای هر ترم، شامل سال، نام ترم، لیست تمام دروس و نمرات اخذ شده در آن ترم، و معدل همان ترم خواهد بود. این گزارش به ترتیب زمانی مرتب شده است.

- RemoveStudentFromCourse

این پراسیجر به یک استاد اجازه میدهد تا دانشجویی را از یکی از کلاس‌هایش در ترم جاری حذف کند. روند کار به این صورت است:

1. ابتدا، پراسیجر با استفاده از تاریخ روز (GETDATE)، ترم تحصیلی فعال فعلی را پیدا میکند.
2. سپس، بررسی می‌کند که آیا استاد مشخص شده (@instructor\_id) واقعاً آن درس خاص (@course\_id) را در ترم جاری تدریس می‌کند یا خیر. اینکار برای اطمینان از این است که استاد فقط بتواند دانشجویان کلاس‌های خودش را حذف کند.
3. اگر این شرایط برقرار بود، پراسیجر اقدام به حذف رکورد ثبت نام (Enrollment) دانشجو از آن درس میکند.
4. در نهایت، با بررسی @ROWCOUNT (تعداد ردیف‌های تحت تأثیر آخرین دستور)، یک پیام مناسب نمایش می‌دهد که آیا دانشجو با موفقیت حذف شده یا اصلاً چنین ثبت نامی برای حذف وجود نداشته است.

- **EnrollCourse**

این پراسیجر، ثبت نام دانشجو در یک درس را کنترل میکند. قبل از ثبت نام، ابتدا سقف واحد دانشجو را بر اساس معدلش میسنجد. سپس چک میکند که کلاس تداخل زمانی نداشته باشد، تکراری نباشد و از سقف واحد مجاز بیشتر نشود. فقط در صورت تأیید همه این موارد، ثبت نام را نهایی میکند.

- **SetGrade**

این پراسیجر برای ثبت نمره ی دانشجو توسط استاد است. قبل از ثبت نهایی، ابتدا صحت نمره (باید بین ۰ تا ۲۰ باشد)، تطابق استاد با درس، و وضعیت ثبت نام دانشجو را کنترل میکند. تنها در صورتی که تمام این موارد درست باشند، نمره را در سیستم ثبت کرده و در غیر این صورت، خطای مربوطه را اعلام میکند.

- **AssignTA**

این پراسیجر برای انتصاب یک دانشجو به عنوان دستیار آموزشی (TA) توسط استاد درس است. روند کار آن بر اساس چند مرحله اعتبارسنجی کلیدی است:

1. بررسی صلاحیت استاد: ابتدا چک می کند که آیا فردی که می خواهد TA منصوب کند، واقعاً استاد همان کلاس است یا خیر.
  2. بررسی وضعیت دانشجو: وضعیت تحصیلی دانشجوی انتخاب شده را بررسی می کند (باید "فعال" یا "فارغ التحصیل" باشد).
  3. جلوگیری از انتصاب تکراری: مطمئن می شود که این دانشجو قبلاً برای همین کلاس به عنوان TA انتخاب نشده باشد.
- اگر هر سه شرط بالا برقرار باشند، انتصاب با موفقیت انجام میشود؛ در غیر این صورت، پیام خطای مربوطه را نمایش میدهد.

## 1. Functions

- **GetStudentExams**

با دریافت شماره دانشجویی به عنوان ورودی، این تابع به سوابق ثبت نام او نگاه میکند و تمام امتحانات مربوط به درس هایی که دانشجو اخذ کرده را پیدا میکند. خروجی آن یک جدول شامل اطلاعات کلیدی برای هر امتحان است: سال و ترم، نام درس، نوع امتحان (مثلاً میان ترم یا پایان ترم)، تاریخ دقیق، و محل برگزاری (شماره اتاق و نام ساختمان).

- **TermGPA**

این تابع برای محاسبه معدل (GPA) یک دانشجوی مشخص در یک ترم خاص است که با دریافت شماره دانشجویی، سال و ترم، به سوابق تحصیلی او در آن ترم مراجعه میکند. سپس، نمره ی هر درس را در تعداد واحد های آن ضرب کرده و مجموع این مقادیر را بر کل واحد های اخذ شده در آن ترم تقسیم می کند تا معدل را به دست آورد.

- TotalGPA

این تابع با دریافت شماره دانشجویی، به تمام سوابق تحصیلی او مراجعه کرده و معدل کل را بر اساس فرمول میانگین وزنی محاسبه میکند. این تابع معدل کل را فقط بر اساس درس‌های پاس شده حساب میکند.

- TotalPassedCredits

این تابع برای محاسبه مجموع کل واحدهای پاس شده توسط یک دانشجوی مشخص است. با دریافت شماره دانشجویی، تمام درس‌هایی که او اخذ کرده را بررسی میکند و فقط درس‌هایی را انتخاب میکند که نمره آنها ۱۰ یا بالاتر باشد. سپس، تعداد واحد‌های تمام این درس‌های پاس شده را با هم جمع کرده و نتیجه نهایی را به عنوان خروجی برمی‌گرداند.

- RecommendCourses

این تابع، با استفاده از اطلاعات دانشجو مثل رشته، ترم فعلی و معدل، لیستی از درس‌هایی که برای ترم بعد مناسب هستند پیشنهاد میدهد. اول درس‌هایی که پیش‌نیاز هایشان کامل پاس شده‌اند را بررسی میکند، سپس درس‌هایی که خودش پیش‌نیاز درس‌های دیگر است، در اولویت قرار می‌گیرند. نهایتاً بر اساس معدل، سقف مجاز واحد تعیین شده و درس‌ها به ترتیب اهمیت تا آن سقف انتخاب میشوند.

- HasScheduleConflict

این تابع با مقایسه روز و ساعت شروع کلاس جدید با تمام کلاس‌هایی که دانشجو از قبل در همان ترم برداشته، تداخل را تشخیص میدهد. اگر حتی یک کلاس با زمان یکسان پیدا شود، تابع مقدار 1 (تداخل) و در غیر این صورت مقدار 0 را برمی‌گرداند.

- CheckGraduationStatus

این تابع با بررسی دو شرط، وضعیت فارغ التحصیلی دانشجو را مشخص میکند: ابتدا چک می‌کند که آیا دانشجو تمام دروس اجباری رشته‌اش را پاس کرده و سپس بررسی میکند که آیا مجموع واحدهای پاس شده‌اش به حد نصاب لازم (مثلاً ۱۴۰ واحد) رسیده است یا خیر. تابع فقط در صورتی که هر دوی این شرایط به طور همزمان برقرار باشند، مقدار 1 (به معنی واجد شرایط بودن) را برمی‌گرداند.

- IsValidNationalID

این تابع، کد ملی را با اجرای الگوریتم استاندارد آن اعتبارسنجی میکند. این کار را با استفاده از ۹ رقم اول برای محاسبه یک رقم کنترل انجام میدهد و سپس آن را با رقم دهم کد ملی مقایسه میکند. در صورت تطابق این دو، کد معتبر و در غیر این صورت، نامعتبر است.

## توضیحات مربوط به اسکیمای Library

### 1. Triggers

- UpdateBookCopyStatusToBorrowed

بعد از اینکه امانتی ثبت شد، وضعیت کپی را در جدول کپی‌ها به borrowed تغییر میدهد.

- UpdateBookCopyStatusToReserved

بعد از اینکه رکورد رزروی ثبت شد، وضعیت کپی را به reserved تغییر میدهد.

- UpdateReservationToCompleted

بعد از اینکه امانتی ثبت شد، در صورتی که کپی قبلاً توسط شخص رزرو شده باشد، وضعیت رزرو را به completed تغییر میدهد.

- BlockInvalidBorrowing

قبل از اینکه امانتی افزوده شود چک میشود در صورتی که کتاب در وضعیت available باشد یا توسط خود شخص رزرو شده باشد و رزرو کامل نشده باشد امانت اضافه نمیشود، در غیر اینصورت rollback میکند.

- LogLibraryMemberChanges

هر تغییری اعم حذف، اضافه و آپدیت روی جدول LibraryMember را ثبت میکند.

- LogBorrowingChanges

هر تغییری اعم حذف، اضافه و آپدیت روی جدول Borrowing را ثبت میکند.

- LogReservationChanges

• هر تغییری اعم حذف، اضافه و آپدیت روی جدول Reservation را ثبت میکند.

### 2. Stored Procedures

- BorrowBook

برای عضو مشخص و یک کپی مشخص امانت ثبت میکند در صورتی که همه شرایط زیر برقرار باشد:

1. همان کتاب توسط همان شخص یا شخص دیگری امانت برده نشده باشد.

2. عضویت فرد فعال باشد.

3. فرد کمتر از 3 کتاب امانت برده باشد.

(اینکه کپی در وضعیت available باشد یا توسط همین شخص رزرو شده باشد توسط تریگر چک میشود.)

- ReserveBook

برای عضو مشخص و یک کپی مشخص رزرو ثبت میکند در صورتی که همه شرایط زیر برقرار باشد:

1. وضعیت کپی available باشد.

2. عضویت فرد فعال باشد.

3. فرد کمتر از 5 کتاب رزرو کرده باشد.

- ReturnBook

در صورتی که شخص آن کپی را امانت گرفته باشد، کتاب را برمیگرداند. پس وضعیت کپی را available میکند و در صورتی که دیر برگدانده باشد، جریمه آن محاسبه و ثبت میشود.

- MarkOverdueAsLost

وضعیت تمامی کپی هایی که بیشتر از 60 روز از موعد بازگشتشان گذشته است را به lost تغییر میدهد.

- BookSuggestion

(مطابق خواسته های داک پروژه)

### 3. Functions

- CountMemberBorrowedBooks

محاسبه تعداد کپی هایی که یک شخص خاص امانت گرفته ولی پس نیاورده است.

- CountMemberReservations

محاسبه تعداد کپی هایی که یک شخص رزرو کرده و هنوز امانتشان نگرفته است. (در حالت pending هستند).

- CalculateMemberDebt

بدهی یک شخص را حساب میکند.

- GetMemberBorrowedBooks

تاریخچه کتاب هایی که یک شخص امانت گرفته را برمیگرداند.

- GetMemberReservedBooks

تاریخچه کتاب هایی که یک شخص رزرو کرده است را برمیگرداند.

- GetMemberFavoriteCategories

محبوب ترین دسته بندی ها برای یک شخص را بر اساس تکرار آنها در کتاب هایی که امانت گرفته بدست می آورد و حداکثر سه تای اول را برمیگرداند.