

	<p>پروژه درس اصول و طراحی پایگاه داده</p> <p>دکتر ممتازی</p> <p>ترم بهار ۱۴۰۱ - دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر</p> <p>زمان تحویل: ۱۴۰۱/۴/۱۱</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 آیدی تلگرام @mohmdRad یا ایمیل [mhmohammadirad@gmail.com](mailto:mhmohammadirad@gmail.com) در ارتباط باشید.
۲. مهلت تحویل پروژه تا ۸ صبح شنبه ۱۴۰۱/۴/۱۱ می باشد و این تاریخ قابل تمدید نیست.
۳. همه کد-های SQL خود را در یک فایل sql. قرار دهید و این فایل را به همراه فایل-های برنامه رابط کاربری به همراه تمامی کتابخانه-های استفاده شده (به صورتی که روی دستگاه-های دیگر قابل اجرا باشد)، در قالب یک فایل zip با فرمت StudentNumber.zip در سایت بارگذاری نمایید.

در این پروژه می-بایست پایگاه داده-ای برای سامانه مدیریت آموزش الکترونیکی که در تمرین دوم طراحی کردید را پیاده سازی کنید.

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-های لازم برای موجودیت-ها در این پروژه ذکر شده است ولی می-توانید بر اساس 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 تمام توانمندی-های خواسته شده تابع یا روالی مجزا با ورودی-های مناسب تعریف کنید که این توابع یا روالها توسط رابط کاربری فراخوانی می-شوند.

توجه داشته باشید که می-توانید از نرم افزارهای DBMS مانند PostgreSQL، MariaDB یا MySQL برای پیاده-سازی پایگاه داده، و هر یک از زبان-های برنامه-نویسی برای پیاده-سازی رابط کاربری، استفاده کنید.

نکات کلی:

- تمامی موارد خواسته شده باید کاملاً در SQL پیاده-سازی شوند و رابط کاربری فقط نتیجه گرفته شده از پایگاه داده را به صورت دلخواه شما نمایش دهد.
- برای نگهداری رشته-های طولانی می-توانید طول ۵۱۲ را در نظر بگیرید.
- برای نگهداری شماره تلفن، کدملی یا شماره-های دیگر از ویژگی‌های عددی استفاده نکنید زیرا صفر ابتدای آن از دست می رود.
- مواردی که با قید "به صورت خودکار" توصیف شده اند، باید با trigger یا با استفاده از کلید خارجی پیاده

سازی شوند.

• موارد امتیازی با سبز نمایش داده شده-اند.

## توضیحات پیش نیاز پروژه

برای استفاده از سامانه مدیریت آموزش الکترونیکی باید لیستی از دانشجویان، اساتید و موارد مربوط به هر یک، از جمله دروس ارائه شده توسط هر استاد و دروس اخذ شده توسط هر دانشجو داشته باشیم. به این منظور به همراه صورت پروژه یک فایل excel شامل چند صفحه وجود دارد که داده‌های مورد نیاز در آن‌ها قرار دارد.

برای راه اندازی سامانه باید ابتدا جداول مورد نیاز خود را در پایگاه داده ایجاد کنید و سپس داده‌های هر جدول را در آن import کنید. با استفاده از IDE هایی نظیر dataGrip می‌توانید طبق [این توضیحات](#) این کار را راحت‌تر انجام دهید؛ همچنین ساختار پایگاه داده خود را بهتر مشاهده کنید، راحت‌تر پرسمان اجرا کنید و داده‌های هر یک از جداول را مشاهده کنید.

در سامانه آموزشی کاربران می‌توانند یکی از دو نقش اصلی دانشجو و استاد را داشته باشند.

### دانشجو:

در صفحه students از excel داده شده، ویژگی‌های مورد نیاز برای دانشجویان آورده شده که هر یک از این ویژگی‌ها در زیر توضیح داده شده است.

- کد ملی (national\_code): کد ملی هر شخص یک رشته عددی ۱۰ رقمی یکتا می‌باشد. هر کد ملی در **کل سیستم مربوط به فقط یک شخص می‌باشد.**
- شماره دانشجویی (student\_no): شماره دانشجویی یک عدد ۷ رقمی یکتا می‌باشد. این شماره می‌تواند به عنوان شناسه کاربر و به صورت کلید در بقیه جداول استفاده شود و نماینده‌ی دانشجو باشد.
- نام و نام خانوادگی: هم به فارسی و هم انگلیسی داده شده است (name\_fa, name\_en). نام و نام خانوادگی شخص با یک «-» جدا شده‌اند.
- نام پدر (father\_name)
- تاریخ تولد (birth\_date): تاریخ تولد به صورت شمسی داده شده است.
- شماره تماس (mobile)
- رشته تحصیلی (major)

پس از وارد کردن ویژگی‌های بالا به جدول مورد نظرتان برای دانشجویان، باید پرسمان **به روز رسانی (update)** ای بنویسید

که دو ویژگی زیر را با توجه به مقدار ثبت شده در بقیه ستون‌ها محاسبه و مقداردهی کند. در این مورد نوشتن پرسمان(query) کافی است و نیازی به تعریف کردن تابع یا روال نیست.

- گذرواژه: گذرواژه اولیه به صورت ترکیب کدملی و حرف اول نام انگلیسی (به صورت بزرگ) و حرف اول نام خانوادگی انگلیسی (به صورت کوچک) است. مثلاً گذرواژه برای حسین پناهی با کدملی ۰۱۲۳۴۵۶۷۸۹ به صورت 0123456789Hp خواهد بود. البته پس از محاسبه این مقدار، باید آن را به صورت درهم‌ریخته (hashed) در بیاورید و مقدار درهم‌ریخته را در جدول ذخیره کنید. برای این کار می‌توانید از تابع MD5 استفاده کنید. در هنگام ورود فرد به حساب خود نیز باید گذرواژه وارد شده ابتدا درهم‌ریخته شود و سپس با آن‌چه در پایگاه داده ذخیره شده است مقایسه شود.
- پست الکترونیکی(email): پست الکترونیکی نیز به صورت ترکیبی از حرف اول نام به همراه نام خانوادگی است؛ مثلاً پست الکترونیکی برای حسین پناهی [h.panahi@aut.ac.ir](mailto:h.panahi@aut.ac.ir) خواهد بود.

استاد:

در صفحه faculty از excel داده شده، ویژگی‌های مورد نیاز برای اساتید آورده شده است. ویژگی‌های اساتید مشابه دانشجویان است؛ فقط به جای شماره دانشجویی، کد استاد (professor\_no) مطرح است که یک عدد ۵ رقمی یکتا می‌باشد. برای اساتید رشته تحصیلی مطرح نیست ولی ویژگی‌های زیر مشخص شده است.

- گروه آموزشی (department)
  - عنوان (title): عنوان استاد که یکی از مقادیر «استاد»، «استادیار» و «دانش‌یار» می‌باشد.
- \*\*\* دقت کنید که گذرواژه اولیه و پست الکترونیکی برای اساتید باید مشابه دانشجویان مشخص شود.

دروس ارائه شده:

در صفحه courses از excel داده شده، مشخصات دروس ارائه شده آورده شده که هر یک از این ویژگی‌ها در زیر توضیح داده شده است.

- کد درس (course\_id): کد درس یک عدد ۸ رقمی یکتا می‌باشد که کلید اصلی برای درس می‌باشد. از این کد در لیست کلاس‌ها برای مشخص کردن دانشجویان مربوط به هر درس استفاده می‌شود.
- نام درس (course\_name)
- استاد درس (professor\_no): کد استاد تدریس کننده درس

\*\* طبق توضیحات تمرین دوم، course درس‌های مجاز است که از آن نمونه‌هایی (instance) گرفته می‌شود و تحت عنوان section ارائه می‌شود ولی در این پروژه برای سادگی مفاهیم، از این کار صرف نظر شده است و دروس ارائه شده به صورت یک‌تا توسط کدشان مشخص می‌شوند.

#### دروس اخذ شده:

در صفحه classrooms از excel داده شده، مشخصات دروس اخذ شده آورده شده که هر یک از این ویژگی‌ها در زیر توضیح داده شده است.

- شماره دانشجویی (student\_no): شماره دانشجویی شخصی که درس را اخذ کرده است.
- کد درس (course\_id): کد درس اخذ شده توسط آن دانشجو

\*\* علاوه بر ویژگی‌های ذکر شده برای هر یک از موارد بالا، ممکن است ویژگی‌های دیگری نیز لازم باشد یا پیاده‌سازی قابلیت‌های مورد نیازتان را ساده‌تر کند. به تشخیص خودتان می‌توانید ویژگی‌های دیگری به هر یک از جداول اضافه کنید.

## قابلیت‌های مشترک

برای هر یک از کاربران باید قابلیت‌هایی فراهم شود. برخی از قابلیت‌ها برای دانشجویان و اساتید کاملاً متفاوت است مثلاً اساتید می‌توانند آزمون تعریف کنند در حالی که برای دانشجویان چنین قابلیتی در نظر گرفته نشده است.

ولی برخی از قابلیت‌ها برای همه کاربران مشترک است. مثلاً همه کاربران باید بتوانند:

- وارد حساب کاربری خود شوند و از آن خارج شوند.

- گذرواژه خود را تغییر دهند.

- دروس مربوط به خود را مشاهده کنند.

## ورود به حساب

کاربران برای ورود باید نام کاربری (شماره دانشجویی یا کد استاد) و گذرواژه خود را در رابط کاربری وارد کنند. در SQL بررسی می‌شود که آیا درهم‌ریخته‌ی گذرواژه‌ی وارد شده با مقدار درهم‌ریخته‌ی نگه‌داری شده برای حساب کاربر برابر است یا خیر. اگر برابر نبودند پیغام خطای مناسب باز می‌گردد و اگر برابر بودند ورود کاربر با موفقیت انجام می‌شود.

ورود کاربر باید در جایی ذخیره شود تا برای درخواست‌های بعدی مشخص باشد که کدام کاربر این درخواست را دارد. برای این کار می‌توانید جدولی در نظر بگیرید که در صورت وارد کردن صحیح نام کاربری و گذرواژه، اطلاعات مورد نیاز از کاربر وارد شده و **زمان ورود را ذخیره کند**. به این صورت برای انجام بقیه‌ی عملیات‌ها نیازی به دوباره فرستادن اطلاعات ورود نیست و آخرین کاربر وارد شده به سیستم به عنوان درخواست دهنده‌ی آن عملیات‌ها در نظر گرفته می‌شود؛ پس وارد کردن نام کاربری و گذرواژه تنها باید یک‌بار و در هنگام ورود کاربر به حساب خود انجام شود.

هنگام خروج کاربر از حسابش نیز باید این ورود به نحوی منقضی شود.

**دقت کنید که مشخص کردن کاربر درخواست دهنده نباید از سمت رابط کاربری انجام شود و باید به صورت کامل در SQL**

**باشد.**

## تغییر گذرواژه

پس از ورود به حساب، کاربر باید بتواند در صورت نیاز گذرواژه خود را تغییر دهد. برای این کار گذرواژه جدید مورد نظرش را وارد می‌کند. ابتدا باید بررسی شود که گذرواژه دو شرط زیر را داشته باشد.

- حداقل ۸ کاراکتر و حداکثر ۲۰ کاراکتر داشته باشد.
- دربردارنده حروف الفبای انگلیسی و عدد (هر دو) باشد.

پس هر گذرواژه باید هم شامل حروف الفبا و هم اعداد باشد و طول گذرواژه نیز کمتر از ۸ کاراکتر و بیشتر از ۲۰ کاراکتر نباشد. در ضمن گذرواژه حساس به حروف کوچک و بزرگ (case sensitive) خواهد بود زیرا در hash کردن تفاوت ایجاد خواهد کرد.

چنانچه هر یک از این شروط برقرار نبود باید پیامی مناسب با آن خطا در SQL برگردانده شود و این پیام در رابط کاربری نمایش داده شود. همچنین گذرواژه تغییر نخواهد کرد.

چنانچه این شروط برقرار بود و مشکلی وجود نداشت، گذرواژه تغییر خواهد کرد و باید درهم‌ریخته‌ی گذرواژه جدید در پایگاه داده ذخیره شود.

## مشاهده دروس

پس از ورود به حساب، کاربر باید بتواند دروس مربوط به خود را مشاهده کند. دروس مربوط به کاربر، برای اساتید دروسی است که ارائه می‌دهند و برای دانشجویان دروسی است که اخذ کرده‌اند.

پس از مشاهده دروس، هر یک از درس‌ها باید قابل انتخاب باشند و کاربر پس از انتخاب آن درس بتواند قابلیت‌های اختصاصی که با توجه به نقشش در نظر گرفته شده است را انجام دهد.

## قابلیت‌های اساتید

اساتید با انتخاب هر درس باید بتوانند:

- لیستی از کل دانشجویان کلاس را مشاهده کنند.
- لیست همه‌ی آزمون‌ها و تمرین‌های درس را مشاهده کنند.
- آزمون و تمرین جدید ایجاد کنند.
- با انتخاب هر یک از آزمون‌ها یا تمرین‌ها همه‌ی پاسخ‌های داده شده به آن را مشاهده کنند.
- به هر یک از پاسخ تمرین‌ها نمره بدهند.

## ایجاد آزمون

برای ایجاد آزمون جدید باید موارد زیر توسط استاد مشخص شود.

- نام آزمون
- زمان شروع آزمون (روز و ساعت)
- زمان پایان آزمون (روز و ساعت)
- طول آزمون (بر حسب دقیقه)
- مجموعه‌ای از سوالات ۴ گزینه‌ای: برای ایجاد هر سوال ۴ گزینه‌ای باید موارد زیر مشخص باشد.
  - متن سوال
  - گزینه‌ها
  - گزینه صحیح



بدیهی است برای این که مشخص باشد هر سوال مربوط به کدام آزمون است و همچنین هر آزمون مربوط به کدام درس است، باید از شناسه‌ها و کلیدهای مناسبی در هر یک استفاده شود.

پس از زمان پایان آزمون نمره آزمون‌ها باید به صورت خودکار (با فعال‌سازی توسط trigger) برای همه دانشجویان محاسبه شود. نمره آزمون به این صورت محاسبه می‌شود که اگر پاسخ دانشجو با پاسخ صحیح سوال یکسان باشد یک نمره و در غیر این صورت صفر نمره در نظر گرفته می‌شود و مجموع این نمرات نمره‌ی آزمون دانشجو را مشخص می‌کند.

### ایجاد تمرین

برای سادگی و نیاز نداشتن به سیستم فایلی، فرض می‌شود تمرین‌ها به صورت مجموعه‌ای از سوالات پاسخ کوتاه هستند که پاسخ‌شان توسط دانشجو در رابط کاربری نوشته می‌شود. برای ایجاد تمرین باید موارد زیر توسط استاد مشخص شود.

- نام تمرین
- مهلت نهایی (deadline)
- مجموعه‌ای از سوالات پاسخ کوتاه: برای هر سوال موارد زیر لازم است.
  - متن سوال
  - پاسخ صحیح

### نمره‌دهی

اساتید باید بتوانند لیستی از همه آزمون‌ها و تمرینات درس را مشاهده و از بین آن‌ها انتخاب کنند.

با انتخاب هر یک از آزمون‌ها باید لیست نمره‌ی همه‌ی افرادی که در آن آزمون شرکت کرده‌اند به استاد نمایش داده شود و همانطور که گفته شد آزمون‌ها با رسیدن به زمان پایان آزمون به صورت خودکار بررسی و نمره‌دهی خواهند شد و نیازی به وارد شدن نمره توسط استاد ندارد.

ولی تمرینات باید توسط استاد نمره‌دهی شود. برای نمره‌دهی به تمرین‌ها، استاد درس باید پس از deadline تمرین، پاسخ‌های دانشجویان را مشاهده کند و نمره مورد نظرش را وارد کند. دقت کنید که اگر استاد قبل از deadline اقدام به نمره‌دهی کرد باید با خطای مناسبی از این کار جلوگیری شود.

## قابلیت‌های دانشجویان

دانشجویان نیز با انتخاب هر درس باید بتوانند:

- لیست همه‌ی آزمون‌ها و تمرین‌های درس را مشاهده کنند.
- در زمان مجاز وارد آزمون شوند و به آن پاسخ دهند.
- به تمرین‌ها پاسخ دهند و پاسخ‌های وارد شده‌شان را مشاهده کنند.

## شرکت در آزمون

دانشجویان باید بتوانند پس از زمان شروع آزمون وارد آزمون شوند و به سؤالات پاسخ دهند. با وارد کردن پاسخ مورد نظرشان برای هر سؤال باید سؤال بعدی نمایش داده شود تا به انتهای سؤالات یا انتهای زمان آزمون برسند.

یعنی هنگام وارد کردن پاسخ دانشجو به هر سؤال در جداول و همچنین نمایش سؤال بعدی، باید ابتدا بررسی شود:

- از زمان پایان آزمون نگذشته باشیم.
- زمانی که دانشجو در حال پاسخ‌دهی به آزمون بوده از طول آزمون بیشتر نباشد. (برای محاسبه زمان پاسخ‌دهی دانشجو می‌توانید از تفاضل زمان حال و زمانی که یک نمونه (instance) از آزمون برای دانشجو ایجاد شده استفاده کنید)

در صورتی که هر یک از این دو شرط برقرار نبود باید خطای مناسبی نمایش داده شود، پاسخ سؤالی که پس از پایان آزمون داده شده ثبت نشود و سؤال‌های بعدی نمایش داده نشود ولی پاسخ سؤالات قبلی ثبت شده باشد.

شرکت در هر آزمون برای هر دانشجو فقط یک مرتبه مجاز است یعنی اگر دانشجویی در آزمون شرکت کرد نباید بتواند دوباره وارد آزمون شود و به سؤالات پاسخ دهد.

دانشجویان باید بتوانند پس از زمان پایان آزمون نمره خود را مشاهده کنند و مرور آزمون داشته باشند یعنی سؤالات آزمون را به همراه پاسخ خودشان و پاسخ صحیح سوال مشاهده کنند.

## پاسخ‌دهی به تمرینات

دانشجویان باید بتوانند تا قبل از deadline تعریف شده برای تمرین وارد تمرین شوند و به سؤالات پاسخ دهند. با وارد کردن پاسخ مورد نظرشان برای هر سؤال باید سؤال بعدی نمایش داده شود تا به انتهای سؤالات برسند.

دانشجو تا قبل از deadline می‌تواند چندین مرتبه وارد تمرین شود و به هر سؤال پاسخ دهد. در صورتی که تمرین مجدداً پاسخ داده شود، باید پاسخ‌های جدید به‌روزرسانی شود و همچنین زمان بارگذاری تمرین نیز به‌روزرسانی شود.

## تدریس‌یار

اضافه کردن تدریس‌یار کلاً امتیازی است.

اساتید می‌توانند برای هر یک از دروسی که ارائه می‌دهند تدریس‌یار انتخاب کنند. تدریس‌یار دانشجویی می‌باشد که در درسی که نقش تدریس‌یار پیدا کرده برخی از قابلیت‌های استاد درس را کسب می‌کند. برای اضافه کردن تدریس‌یار، استاد درس باید شماره دانشجویی شخص مورد نظرش را وارد کند. شماره دانشجویی مورد نظر باید بررسی شود که

- شماره دانشجویی معتبری باشد و چنین شماره دانشجویی وجود داشته باشد.
- دانشجوی مورد نظر جزو دانشجویان همین درس نباشد یعنی یک شخص نمی‌تواند هم تدریس‌یار درس باشد هم دانشجوی درس.

چنانچه هر یک از این موارد برقرار نبود باید خطای مناسب در SQL بازگردانده شود و در غیر این صورت شماره دانشجویی شخص مورد نظر به عنوان تدریس‌یار این درس ثبت شود.

تدریس‌یار علاوه بر قابلیت‌های مشترک و قابلیت‌های دانشجو که بالاتر ذکر شد و به خاطر دانشجو بودنش دارد، در مورد درسی که تدریس‌یاری آن را بر عهده دارد قابلیت‌هایی شبیه به استاد خواهد داشت.

هنگام نمایش دروس مربوط به دانشجو، علاوه بر نمایش دروسی که شخص آن‌ها را اخذ کرده است و با انتخاب آن‌ها قابلیت‌های دانشجو را دارد، باید دروسی که شخص تدریس‌یار آن است نیز نمایش داده شود.

تدریس‌یار با انتخاب این دروس باید بتواند:

- لیستی از کل دانشجویان کلاس را مشاهده کند.
- لیست همه‌ی تمرین‌های درس را مشاهده کند(نباید آزمون‌ها نمایش داده شود).
- تمرین جدید ایجاد کنند.
- با انتخاب هر یک از تمرین‌ها همه‌ی پاسخ‌های داده شده به آن را مشاهده کنند.
- به هر یک از پاسخ تمرین‌ها نمره بدهند.

همان‌طور که مشاهده می‌شود تدریس‌یار فقط در مورد تمرینات دسترسی دارد و به آزمون‌ها دسترسی ندارد. توضیحات لازم برای هر یک از موارد بالا مشابه قابلیت‌هایی است که بالاتر برای اساتید توضیح داده شد.

**\*\* مجدداً توجه شود که همه قابلیت‌های خواسته شده (به غیر از ورود داده‌های excel به جداول و تولید گذرواژه اولیه و پست الکترونیکی با توجه به آن‌ها) باید در تابع (function) یا روال (procedure) پیاده‌سازی شود و این توابع و روال‌ها در مکان مورد نیاز توسط رابط کاربری با دریافت ورودی‌های لازم فراخوانی شوند.**

## کار با رابط کاربری

دانشجویان نیز باید بتوانند لیستی از دروسی که اخذ کرده‌اند را مشاهده کنند. برای هر درس نیز باید فعالیت‌های زیر را بتوانند انجام دهند.

- برای پیاده‌سازی رابط کاربری می‌توانید از زبان برنامه‌نویسی یا اسکریپت دلخواه خود استفاده کنید.
- تنها کار رابط کاربری تفسیر ورودی‌های کاربر و فراخوانی تابع‌های SQL با ورودی‌های مناسب و ساماندهی (در صورت نیاز) و نمایش خروجی این تابع‌ها می‌باشد.
- برای رابط کاربری، پیاده‌سازی برنامه‌ای که با کنسول ورودی و خروجی را بگیرد کافی است.
- خروجی باید به صورت جدول باشد به صورتی که نام ستون‌ها (ویژگی‌ها) و مقدار آن‌ها مشخص باشد. چه در GUI و چه در CLI

## نکات پایانی

- افرادی که رابط کاربری را **پیاده‌پیاده‌سازی** نمی‌کنند و تنها بخش SQL را انجام می‌دهند ۸۰ درصد نمره را خواهند گرفت.
- پیاده‌سازی رابط کاربری به صورت GUI **حداکثر ۵ درصد** نمره امتیازی خواهد داشت.
- افرادی که فقط رابط کاربری را پیاده‌سازی کنند، نمره‌ای دریافت نخواهند کرد.
- منطق سامانه نباید در رابط کاربری پیاده‌سازی شود..

موفق باشید