



* فاز دوم و سوم پروژه SQL

* درس: پایگاه داده * استاد: دکتر فرضی

* موعد بارگذاری فاز دوم: ۱۴۰۱/۱۰/۰۹

* موعد بارگذاری فاز سوم: ۱۴۰۱/۱۰/۳۰

✓ فاز دوم این پروژه شامل دو بخش است، بخش اول مربوط به نوشتن دستورات SQL و بخش دوم مربوط به اتصال پایگاه داده به برنامه‌ای است که تنها شامل لایه Data باشد.

✓ فاز سوم این پروژه شامل پیاده‌سازی لایه Business و Presentation (یا UI) بوده و انجام آن اختیاری است.

توجه فرمایید شما باید روی پایگاه داده‌ای که پیش‌تر در فاز اول پروژه طراحی نموده‌اید تغییرات لازم را اعمال کنید بطوریکه ضمن اصلاح موارد در نظر گرفته نشده، همه جدول‌ها نیز در سطح سوم نرمال‌سازی قرار گیرند. دقت کنید که برای تغییر دادن جدول‌ها، باید از دستورات SQL همچون drop، alter و ... استفاده کنید.

فاز دوم – بخش اول (SQL):

❖ خروجی این فاز شامل دو نوع فایل است:

- فایل‌هایی با پسوند *sql* که دستورات در آنها قرار دارند.
- یک فایل *pdf* شامل نمودار گرافیکی فاز قبل و نمودار نهایی (به دست آمده در این فاز) که باید با یکدیگر بطور خلاصه مقایسه کنید. همچنین تمام خروجی‌های حاصل از اجرای دستورات SQL را در این فایل قرار دهید.

❖ دستورات alter و drop و ...:

- کلیه اصلاحاتی که لازم است روی جدول‌های خود اعمال کنید تا بتوانید دستورات این بخش را اجرا نمایید، با استفاده از دستورات SQL پیاده‌سازی کنید.
- دقت کنید که مشکلات جزئی موجود در فاز قبل (برای تمام گروه‌ها) باید برطرف گردد.
- این مشکلات شامل در نظر نگرفتن بسیاری از جداول، استفاده نادرست از ارث‌بری و جدول‌های میانی و نیز نرمال نبودن جدول‌ها است.

❖ دستورات Update و Insert:

۱. تعداد ۶ کاربر با مشخصات مختلف به پایگاه داده افزوده شود. ✓
- (زهرة رسولی، الهام نیایشی، رضا ملکی، مرتضی مولایی، امید احمدی، محمد سجادی)
۲. ۵ نوع بیمه زلزله ایجاد کنید و آن‌ها را بر اساس بیشترین امتیاز کاربران مرتب کنید. ✓
۳. یک نوع بیمه آتش‌سوزی برای شرکتی دلخواه (مانند بیمه پاسارگاد) به پایگاه داده اضافه کنید. ✓
۴. برای کاربر محمد سجادی، یک بیمه درمان تکمیلی انفرادی (دلخواه) سفارش دهید. ✓
۵. کد معرف رضا ملکی را برای امید احمدی وارد کنید. از تعداد دعوت باقی‌مانده رضا ملکی یک عدد کم کنید. ✓
۶. بیمه درمان تکمیلی دی از طرح شخص واحد - سه ستاره را به پایگاه داده خود اضافه کرده و روی آن تخفیف ۶۰ درصد را اعمال کنید. ✓
۷. برای کاربر مرتضی مولایی دو یادآور را برای تاریخ‌های ۱۴۰۱/۱۰/۱۰ و ۱۴۰۱/۱۱/۱۱ ثبت کنید. ✓

❖ دستور Delete:

✓ کاربرانی را که تاکنون هیچ بیمه‌ای را سفارش نداده‌اند، حذف کنید.

❖ دستور Insert:

✓ جدول جدیدی بسازید و مشخصات تمام کاربرانی که بیمه آتش‌سوزی را از استان‌های خراسان شمالی، خراسان رضوی و خراسان جنوبی سفارش داده‌اند در آن درج نمایید.

❖ دستور View:

لیست کاربرانی که بیمه درمان تکمیلی خانوادگی را سفارش داده‌اند، نمایش دهید.

پردازه‌های ذخیره شده (Stored Procedure):

پرسمان‌های زیر را الزاماً بصورت پردازه ذخیره شده بنویسید.

برای هر پردازه از نام مشخص شده استفاده کنید. پردازه‌ها را بصورت پارامتری بنویسید و سپس با مقادیر مشخص شده اجرا کنید. توجه کنید خروجی هیچ پرسمانی نباید تهی باشد^۱.

^۱ در این صورت، داده‌های موجود در جدول‌ها را افزایش دهید.

۱. نام پردازش ذخیره شده: **userNotOrder**

نمایش نام خانوادگی کاربرانی که یا تا به حال هیچ بیمه عمری سفارش نداده‌اند.

۲. نام پردازش ذخیره شده: **userMaxInsurance**

نمایش نام و نام خانوادگی کاربرانی که بیمه درمان تکمیلی انفرادی با مبلغ سالانه حداکثر X تومان را سفارش داده‌اند.

$$X = 1000000$$

۳. نام پردازش ذخیره شده: **userReminder**

نمایش نام کاربرانی که یادآوری بیمه عمر ثبت کرده‌اند بطوریکه دوره پرداخت آنها بصورت X است.

$$X = \text{هر سه ماه}$$

۴. نام پردازش ذخیره شده: **avgName**

نمایش میانگین تعداد حروف نام خانوادگی کاربرانی که بیمه درمان تکمیلی انفرادی برای یک نفر با بیمه‌گر پایه X سفارش داده‌اند و حداکثر Y بیمه از سایت از کی سفارش داده‌اند.

$$X = 3 \text{ و } Y = \text{بانک‌ها}$$

۵. نام پردازش ذخیره شده: **nameCompanyAsc**

نمایش لیست پیشنهادی بیمه آتش‌سوزی استان X به ترتیب ارزان‌ترین.

$$X = \text{اصفهان}$$

۶. نام پردازش ذخیره شده: **userInvitation**

نمایش نام کاربرانی که برای حداقل A نفر کد دعوت را ارسال کرده‌اند و حداقل B بیمه سفارش داده‌اند.

$$A = 2 \text{ و } B = 3$$

۷. نام پردازش ذخیره شده: **userCompanyInsurance**

نمایش تمام اطلاعات کاربرانی که بیمه B را از شرکت A سفارش داده‌اند.

$$A = \text{دی} \text{ و } B = \text{عمر}$$

۸. نام پردازش ذخیره شده: **usersMaxMin**

نمایش نام دو کاربری که مجموع بیمه‌های سفارش داده شده توسط آنها تاکنون در میان تمام کاربران کمینه و بیشینه باشد.

۹. نام پردازش ذخیره شده: **mobileCount**

نمایش تعداد شماره‌های موبایلی که با X شروع می‌شوند.

$$X = 0937$$

۱۰. نام پردازش ذخیره شده: **ageRangeFamilyInsurance**

نمایش تعداد و مجموع مبلغ سالانه بیمه‌های درمان تکمیلی خانوادگی X نفره به تفکیک بازه سنی سرپرست آنها.

$$X = 2$$

۱۱. نام پردازش ذخیره شده: **twoInsurances**

نمایش نام و نام خانوادگی کاربرانی که هم بیمه A و هم بیمه B را سفارش داده‌اند.

آتش سوزی = B و درمان تکمیلی = A

۱۲. نام پردازش ذخیره شده: **avgEarthquake**

نمایش میانگین متراژ املاک نوساز استان X به تفکیک هر شهر که برای آنها بیمه زلزله سفارش داده شده است.

$$X = \text{سمنان}$$

۱۳. نام پردازش ذخیره شده: **updateInsurance**

ویرایش (بروزرسانی) روش پرداخت تمام بیمه‌نامه‌های عمری که مدت قرارداد آنها کمتر از X سال است به شش ماهه.

$$X = 7$$

۱۴. نام پردازش ذخیره شده: **DeleteInsurance**

حذف تمام بیمه‌های عمر ۵ ساله با توان پرداخت بین X و Y تومان که مبلغ (کل) ارزش بازخرید آنها کمتر از Z تومان باشد.

$$X = 50000 \text{ و } Y = 100000 \text{ و } Z = 30000000$$

۱۵. نام پردازش ذخیره شده: **fireInsurance**

نمایش نام خانوادگی کاربرانی که یا بیمه آتش‌سوزی را برای تمام انواع ملک سفارش داده‌اند، یا فقط برای یک آپارتمان یا مجتمع با بیش از ۱۰ واحد سفارش داده‌اند.

راهنما:

پردازش ذخیره شده

<https://www.aparat.com/v/V4Gg8>

کار با تقویم شمسی

<https://stackoverflow.com/questions/6706692/how-to-convert-datetime-in-persian-in-sql-server>
<https://raresql.com/2013/05/08/sql-server-how-to-convert-gregorian-dates-to-hijri-date-with-formatting/>

فاز دوم – بخش دوم (اتصال به برنامه):

در نهایت یک برنامه کامل را در نظر بگیرید (یعنی نرم‌افزاری شامل لایه‌های Data، Business/Application و Presentation).

در این فاز تنها لازم است لایه Data را پیاده کنید، به این معنی که در یک زبان برنامه‌نویسی دلخواه با استفاده از استانداردهای اتصال به پایگاه داده (مانند JDBC و ODBC و ADO.NET) برنامه را به پایگاه داده وصل کنید. ابتدا یک interface به صورت زیر در برنامه خود ایجاد کنید:

```
Interface Repository<Entity,ID extends Serializable> {  
    Entity findById(ID id);  
    List<Entity> findByIds(Collections<ID> ids);  
    List<Entity> findAll();  
    Boolean deleteById(ID id);  
    Boolean DeleteByIds(Collection<ID> ids);  
    Entity save(Entity E); // saves the Entity, if already exists updates it, then return  
    the entity with the given id by the database  
    ...  
}
```

برای این کار دو پکیج Model و Repository را در نظر گرفته و به ازای هر entity در پایگاه داده حداقل یک کلاس بسازید. فیلدهای مناسبی در نظر بگیرید و این کلاس‌ها را در پکیج مدل قرار دهید. سپس، برای هر entity، یک کلاس DAO بسازید که واسط Repository را پیاده کند. سپس، متدهای این کلاس‌های DAO را پیاده‌سازی کنید. مثلاً برای یک User entity که id آن از جنس bigint باشد، کلاس User را مطابق زیر بسازید:

`Class UserDao implements Repository<User,Long>`

فاز ۳ (اختیاری / امتیازی):

لایه Business و لایه Presentation را پیاده‌سازی کنید.
این برنامه می‌تواند یک اپ دسکتاپ (با local database) یا وبسایت یا موبایل باشد.
منظور از این دو لایه این است که برنامه شما باید یک UI داشته باشد که از طریق آن بتوان با برنامه کار کرد.
منطق برنامه نیز باید نوشته شود، UI خالی نباشد و برنامه قابل استفاده باشد.

نکات مهم:

- ❖ تحویل فاز دوم و سوم این پروژه به صورت گروهی و در دو مرحله بارگذاری در سایت و تحویل شفاهی (آنلاین) انجام می‌پذیرد. (بارگذاری فایل‌های پروژه توسط تمام اعضای گروه الزامی است).
- ❖ دقت کنید که تمام فایل‌های مربوط به هر دو بخش و خروجی‌های به دست آمده را باید بصورت فشرده شده بارگذاری نمایید.
- در صورت بالا بودن حجم فایل‌ها می‌توانید یک repository مانند gitlab یا github بصورت public ایجاد و لینک آن را بارگذاری کنید.
- ❖ هر گروه در فاز پیشین مشخص شده است. (گروه خود را تغییر ندهید).
- ❖ در صورت عدم حضور برای ارائه شفاهی، نمره‌ای برای فازهای دوم (و یا سوم) این پروژه منظور **نخواهد شد**.
- ❖ زمان تحویل شفاهی (آنلاین) برای فازهای دوم و سوم متعاقباً در کانال تلگرام درس اعلام خواهد شد.
- ❖ در صورت سؤال یا ابهام در مورد این پروژه می‌توانید در گروه تلگرام درس و یا با شناسه‌های زیر مطرح فرمایید.

@HassanTSF، @iammobi78، @SaedehTahery و @SZAftaB

😊 موفق و سربلند باشید 😊