



اصول طراحی پایگاه داده

مدرس: دکتر عین الله خنجری - بهار 1402-1403

دانشگاه علم و صنعت ایران

----- فاز پنجم: Trigger, Assertion, View, Procedure, Function -----

مهلت تحویل: 4 تیر 1403

نحوه تحویل: بارگذاری در کوئرا (توسط هر دو عضو تیم)

انجام در گروه های دو نفره + مشورت با منتور

توضیحات فاز پنجم پروژه:

در این فاز قرار است برخی امکانات پیشرفته SQL را روی پروژه تان اعمال کنید. مثل مرحله قبلی، نحوه اعمال کردن این موارد به دلخواه خودتان است و بسته به نیاز و موضوع پروژه، ایده ها ممکن است متفاوت باشد. برای همین معیار ارزیابی این فاز، نحوه صحیح پیاده سازی موارد خواسته شده و ایده ای است که برای هر کدام به کار خواهید گرفت.

جزئیات مواردی که باید پیاده سازی شوند در زیر آمده است. با این که برای هر کدام توضیح مختصری داده شده، اما احتمالاً نیاز باشد در مورد سینتکس هر کدام و نحوه بکارگیری شان بیشتر جستجو کنید. همچنین خوب است در مورد اینکه هر کدام در چه شرایطی کارآمد هستند هم کمی مطالعه کنید تا در جای درست و با ایده مناسبی در پروژه خودتان از آنها استفاده کنید.

ضمناً در این فاز بیشتر از فازهای قبل احتمال دارد سینتکسی که می زنید، به ازای هر محیطی که در آن از SQL استفاده می کنید متفاوت باشد. پس در این مورد هم دقت لازم رو داشته باشید.

Trigger - 1

تا حالا در مورد اثر پروانه ای (Butterfly Effect) چیزی شنیده اید؟ اثر پروانه ای یعنی به عنوان مثال پروانه ای در آمریکای جنوبی پر بزند و این پر زدن به طور زنجیره وار موجب اتفاقاتی شود که در نهایت به وقوع زلزله ای در آسیای شرقی بیانجامد! Trigger ها در دیتابیس مثل اثر پروانه ای می مانند. Trigger را می نویسیم تا با اتفاقی مثل insert یا delete شدن یک داده، خود به خود تغییری که می خواهیم در جایی دیگر اعمال شود. مثلاً یک فیلد عددی، یک شماره زیاد و کم شود یا تغییر های دیگر... Trigger در دیتابیس های امروزی کاربرد زیادی دارد و قرار است در این فاز، شش Trigger برای پایگاه داده پروژه تان بنویسید.

2 - Function

توابع (function ها) در پایگاه داده، مثل توابع در جاهای دیگر، پارامتر هایی را به عنوان ورودی می گیرند و چیزی هم به عنوان خروجی بر می گردانند. محیطی که در آن با SQL کار می کنید به احتمال زیاد توابع آماده زیادی برای استفاده شما دارد و ممکن است تا الان از چند تای آنها استفاده کرده باشید. اما در این بخش قرار است خودتان **function** جدید بنویسید و آن ها را در پروژه تان به کار بگیرید. توابعی که می نویسید باید کار معنی داری در context پروژه تان انجام دهد. استفاده از توابع آماده در بدنه تابع تان مجاز است.

Procedure -3

یک Procedure در SQL را اصولا زمانی می نویسیم و ذخیره اش می کنیم که بخواهیم یک یا چند عمل متوالی متوالی را چند بار انجام دهیم. مثلا بجای اینکه یک کوئری پر کاربرد را چندین بار اجرا کنیم و نتیجه اش را ببینیم، بجای آن، کوئری مذکور را در یک Procedure می نویسیم و هرگاه خواستیم آن Procedure را اجرا می کنیم. در بنده Procedure مجاز به استفاده از function ها و procedure های دیگر هم هستیم. ضمنا Procedure می تواند یک یا چند input parameter داشته باشد. در این فاز قرار است **Procedure** بنویسید که کار معنی داری در پایگاه داده شما انجام دهند. ضمنا **حداقل یکی شان باید input parameter داشته باشد.**

View -4

گاهی در دیتابیس یک شرکت بزرگ که داده هایمان زیاد و real time هستند، نیاز می شود جدولی را به طور پیوسته مانیتور کنیم و تغییرات آن را به طور زنده تماشا کنیم. View در SQL به همین منظور شکل گرفته است. با درست کردن view از یک جدول یا کوئری دلخواه، قادر به مانیتور کرده زنده آن جدول خواهیم بود. در table های عادی برای دیدن نتایج جدید باید به روزرسانی شان کنیم و به عبارتی دوباره از دیسک آن جدول را بخوانیم. اما view ها اصولا روی RAM ذخیره میشوند تا تغییرات به صورت سریع و لحظه ای به ما نشان داده شود. در پایگاه داده پروژه تان **View** بسازید.

Assertion -5

هنگام ساختن یک جدول با دستور create table بعد از تعریف ستون ها و دیتا تایپ هایشان، محدودیت (Constraint) هایی مثل Foreign key یا not null می نویسیم یا اینکه با استفاده از check قاعده ای تعریف می کنیم که همواره روی داده هایمان صدق کند. این محدودیت ها را به جز در دستور create table یا Alter table ، در Assertion ها هم می توان نوشت. مثلا در یک Assertion چک می کنیم که حقوق یک کارمند از میانگین حقوق رئیس هایش کمتر باشد و اگر حقوق یک فرد را اضافه کنیم به طوری که این شرط برای یک یا چند کارمند نقض شود، آن وقت به ما هشدار داده می شود و این تغییر انجام نمی گیرد. Assertion ها فقط مربوط به جدولی خاص نیستند و روی کل دیتابیس نوشته می شوند و این مورد آنها را از Trigger یا مواردی که کاری مشابه انجام می دهند، متمایز می کند. **Assertion** مناسب و معنی دار در پروژه تان ایجاد کنید.