

حدیث احمدیان 9622613 *** فاز سوم پروژه آزمایشگاه دیتابیس

در این فایل توضیحات مربوط به فاز سوم را مشاهده میکنید، فایل دیگری درمورد موجودیت ها و خود دیتابیس ضمیمه شده.

تغییرات نهایی جداول :

یک) فیلد پسونرد مجددا به جدول مشتری اضافه شد --> دلیل : برای امنیت تهیه ی بلیط و محفوظ ماندن اطلاعات مشتری ها بهتر است هر مشتری برای حساب کاربری خود پسونرد داشته باشد و برای هر بار کار با سیستم به منظور لاگین از پسونرد خود استفاده کند تا احراز هویت صورت گیرد.

```
ALTER TABLE Customer ADD PassWord varchar(50) NOT NULL default 'default_pass'
```

دو) جدولی برای لاگ گرفتن به مجموعه جداول اضافه شد که دارای یه فیلد برای ثبت لاگ هایی است که بعدا توضیح داده میشود:

```
CREATE TABLE my_logs (description_of_log varchar(200) NOT NULL);
```

توضیحات View ها :

singer_sale ← خروجی : تعداد بلیط های فروخته شده توسط هر خواننده را بر میگرداند

	Lana	Adele	Eminem	SIA	Billie
1	7	6	2	2	3

کاربرد : بررسی موفقیت و فروش خواننده ها برای تصمیم گیری های آینده

concert_analysis ← خروجی :دسته بندی و شمارش کنسرت ها بر اساس ژانر و مکان

	genre	location	count
1	Alternative	Walt Disney Concert Hall	1
2	Alternative	All location	1
3	Pop	Royal Albert Hall	1
4	Pop	The Copenhagen Concert Hall	1
5	Pop	The Oslo Opera House	1
6	Pop	All location	3
7	Rapper	National Centre for the Performing Arts	1
8	Rapper	All location	1
9	All genre	All location	5

کاربرد :گزارشی از فعالیت مجموعه برای تحلیل و تصمیم گیری

customer_analysis ← نمایش دسته بندی شده ی بلیط ها ی خریداری شده براساس کلاس آنها، شمارش تعداد کل بلیط و محاسبه قیمت بلیط های خریداری شده به ازای هر مشتری

	customer row	customer tickt row	CustomerName	CustomerFamily	Class	price
1	1	1	amirhosein	sadeqi	1	1500.00
2	1	2	amirhosein	sadeqi	2	2000.00
3	2	1	Faride	Ahmadian	1	1500.00
4	2	2	Faride	Ahmadian	1	3000.00
5	2	3	Faride	Ahmadian	1	4500.00
6	2	4	Faride	Ahmadian	1	6000.00
7	2	5	Faride	Ahmadian	2	6600.00
8	2	6	Faride	Ahmadian	2	7100.00
9	2	7	Faride	Ahmadian	3	7510.00
10	2	8	Faride	Ahmadian	3	7910.00
11	3	1	nima	sabahi	1	2000.00
12	3	2	nima	sabahi	3	2400.00
13	4	1	Parisa	qaderi	1	1500.00
14	4	2	Parisa	qaderi	1	3100.00
15	4	3	Parisa	qaderi	3	3630.00
16	5	1	Parmida	Salimi	1	2000.00
17	5	2	Parmida	Salimi	1	3000.00

کاربرد : تحلیل مشتری مثلا برای پیشنهاداتی که سیستم برای خرید بلیط به مشتری خواهد داد.

all_concerts_data ← نمایش جامع اطلاعات کنسرت ها (با استفاده از جوبین)

	Name	artist	location	genre	Date
1	Born To die World tour	Lana	Walt Disney Concert Hall	Alternative	2012-05-06 20:00:00.000
2	2015 world tour	Adele	Royal Albert Hall	Pop	2015-05-08 20:00:00.000
3	LSD	SIA	The Copenhagen Concert Hall	Pop	2018-01-06 10:00:00.000
4	where do we go tour	Billie	The Oslo Opera House	Pop	2019-12-27 15:00:00.000
5	2020 state tour	Eminem	National Centre for the Performing Arts	Rapper	2020-02-03 12:30:00.000
6	dreams	Lana	The Copenhagen Concert Hall	Alternative	2021-05-06 20:00:00.000

کاربرد : نمایش اطلاعات کنسرت ها به صورت یکجا و جامع (به جای جداول پراکنده) به مشتری برای بررسی و انتخاب کنسرت ها.

dbo.order_price ← ورودی : آیدی سفارش (primary key از جدول Customer_Order)

خروجی : محاسبه ی قیمت نهایی یعنی جمع بلیط های درون سفارش با اعمال تخفیف نهایی

(No column name)	
1	2160.00

نمونه خروجی (ورودی ۵) :

کاربرد : محاسبه قیمت نهایی هر سفارش

dbo.mail_domain ← ورودی : آیدی مشتری (primary key از جدول Customer)

خروجی : دامنه ی ایمیل مشتری

(No column name)	
1	gmail.com

نمونه خروجی (ورودی ۵) :

کاربرد : بررسی معتبر بودن ایمیل وارد شده هنگام ثبت نام

dbo.TicketCount ← ورودی : آیدی دو لوکیشن (primary key از جدول location)

خروجی : آیدی های بلیط های خریداری شده که کنسرت آنها در لوکیشن اول بوده ولی در لوکیشن دوم نبوده

	ID	count
1	1	-
2	13	-
3	17	-
4	2	-
5	3	-
6	4	-
7	5	-
8	Count Of All	7

نمونه خروجی (ورودی ۱ و ۲) :

کاربرد : مقایسه محبوبیت و میزان فروش دو سالن برای تصمیم گیری های آینده

available_ticket ← ورودی : آیدی کنسرت (primary key از جدول concert)

خروجی : آیدی بلیط های کنسرت مورد نظر که هنوز فروخته نشده اند.

	ID
1	21
2	22

نمونه خروجی : (ورودی 1)

کاربرد : اینکه مشتری ببیند چه بلیط هایی برای کنسرت مورد نظرش در دسترس است تا در جدول بلیط ها اطلاعات آنرا بررسی کند و برای خریدش تصمیم گیری کند.

توضیحات procedure ها :

دلیل اینکه موارد زیر به صورت روبه های ذخیره شده نوشته شده اند این است که کاربرد زیادی دارند و مکررا استفاده میشوند پس بهینه است در حافظه ذخیره شده باشند .

SignUp ← ورودی (نام ، نام خانوادگی، ایمیل ، جنسیت، نام کاربری (کلید پرایمری جدول مشتری)، و پسوندد)

این روبه در صورتی که نام کاربری (ID) وارد شده توسط کاربر تکراری نباشد اطلاعات او را در سطر جدیدی وارد جدول مشتری ها می کند و در واقع او را ثبت نام میکند. این روبه خروجی ندارد.

Login ← ورودی (نام کاربری (کلید پرایمری جدول مشتری)، و پسوندد) اگر پسوندد و نام کاربری وارد شده صحیح باشد یعنی رکورد نظیر آن در جدول یافت شود روبه 1 بر میگردد در غیر این صورت 0 بر میگردد.

change_pass ← (نام کاربری (کلید پرایمری جدول مشتری)، و پسوندد، پسوندد جدید) اگر پسوندد و نام کاربری وارد شده صحیح باشد یعنی رکورد نظیر آن در جدول یافت شود رکورد نظیر آن در جدول آپدیت میشود و پسوندد جدید به جای پسوندد قبلی قرار میگیرد و اگر پسوندد صحیح نباشد پیام خطای 'WRONG PASSWORD' چاپ می شود.

create_order ← ورودی (نام کاربری (کلید پرایمری جدول مشتری)) اگر نام کاربری وارد شده درست باشد و وجود داشته باشد یک سفارش جدید برای آن کاربر ساخته میشود. آیدی سفارش (کلید پرایمری جدول order_customer) به طور خودکار توسط روبه محاسبه میشود و مقدار تخفیف به طور پیش فرض 0 در نظر گرفته میشود هم چنین زمان فعلی سیستم به عنوان زمان سفارش ثبت میشود. بعدا کاربر میتواند بلیط های مختلفی به این سفارشش اضافه کند. اگر هم نام کاربری درخواست کننده اشتباه باشد پیام خطای "wrong customer" چاپ می شود.

add_to_order ← ورودی (شماره سفارش (کلید پرایمری جدول order_customer) و شماره بلیط (کلید پرایمری جدول ticket)) این روبه بلیطی که آیدی آن داده شده را به سفارش مورد نظر اضافه میکند (هر سفارش میتواند شامل چند بلیط باشد) در صورتی که درخواست کننده شماره سفارش یا شماره بلیط اشتباهی را داده باشد یا بلیط از قبل فروش رفته باشد پیام خطایی به صورت مقابل چاپ خواهد شد 'wrong order or ticket'

توجه : چون خروجی روبه ها جدول نبود ، اسکرین شاتی از خروجی قرار ندادم اما برای همه ی روبه ها و کلا سایر موارد تست هایی در فایل sql قرار داده شده که با اجرای آنها میتوانید خروجی و یا نتیجه را بررسی کنید.

: anti_DOS

یکی از حملات رایج در سیستم های فروش آنلاین حمله ی (DOS) denial of service یا انکار دریافت سرویس است که در آن شخص مهاجم سفارشی را ثبت میکند و سپس منکر آن میشود و به این شکل به شخص سرویس دهنده خسارت وارد میکند (چون سرویس دهنده برای آماده کردن سفارش ها باید منابع خود را صرف کند، مهاجم تعداد زیادی سفارش ثبت کند و سپس منکر آن شود و به این شکل طعمه خسارت ببیند) حال این تریگر به شکل ساده ای در تلاش است تا جلوی این حمله را بگیرد به این شکل که اجازه نمیدهد رکوردی از جدول سفارش ها (Customer_Order) حذف شود و در صورت تلاش برای حذف آن پیام خطایی به شکل مقابل چاپ خواهد شد 'NO ACCESS TO RETURN ORDER'

برای بررسی خروجی این مورد هم میتوانید به تست ها مراجعه کنید.

: pass_change

به طور کلی نگهداری یک فایل لاگ برای ذخیره ی اطلاعات مهم در دیتابیس حایز اهمیت است که به همین منظور جدول my_logs ساخته شد ، حال این تریگر در هر بار تغییر پسورد یه رکورد در جدول لاگ ذخیره میکند که در آن ثبت میشود کدام کاربر از چه پسوردی به چه پسوردی تغییر پیدا کرد است. کاربرد آن بررسی موارد مشکوک در حمله به اکانت های دیتابیس است. نمونه :

	description_of_log
1	password changed for user 5 from : hello to : hello
2	password changed for user 7 from : new to : very new
3	concert : dreams with id : 6 has been modified in Jul 19 2020 8:57PM

: concert_addition

مورد دیگری که ثبت آن در جدول لاگ میتواند مورد استفاده قرار بگیرد ایجاد کنسرت جدید است . کاربرد آن میتواند بررسی تاریخچه ی ثبت کنسرت ها باشد یا اینکه هربار رکوردی به این شکل اضافه شد به کاربران نوتیفیکیشنی مبتنی بر برگزاری کنسرت جدید ارسال شود. در این تریگر نام کنسرت ، آیدی آن و زمان ایجاد آن در لاگ ثبت میشود. نمونه :

	description_of_log
1	password changed for user 5 from : hello to : hello
2	password changed for user 7 from : new to : very new
3	concert : dreams with id : 6 has been modified in Jul 19 2020 8:57PM

کد مربوط به تمام موارد فوق در فایل sql codes.sql ضمیمه شده است. برای هر مورد نیز تست مناسبی نوشته شده.

بررسی ساختار و اصول دیتابیس :

اول از همه توجه شده است که بین موجودیت ها روابط many to many برقرار نباشد و در صورت وجود موجودیت های واسط جایگزین آن شود مانند جداول (ticket_customer و order_ticket) چون هر مشتری میتواند چند سفارش و هر سفارش چند بلیط داشته باشد. از ابتدای طراحی دیتابیس سعی بر آن بود که دیتابیس نرمال باشد و در بررسی مجدد نیز از این جهت مشکلی دیده نشد، در طی مراحل فقط چند فیلد و جدول به دیتابیس اضافه شد که توضیحات آن در فایل دیتابیس و جداول ضمیمه شده است.

در هیچ جدولی multivalued attribute دیده نمی شود ← فرم اول نرمال

هیچ attribute ی در هیچ جدولی توسط چیز دیگری به جز کلید پرایمری آن جدول به طور کامل تعریف نمیشود ← فرم دوم نرمال

در هیچ جدولی transitive dependency نداریم یعنی attribute های غیر کلید به یکدیگر وابستگی ندارند ← فرم سوم نرمال