

## به نام خدا

### فاز سوم پروژه

### عاطفه کاظمی

#### موضوع پروژه: پایگاه داده ی مربوط به مدیریت هوشمند محوطه ی پارکینگ

از آنجا که امروزه در اکثر نقاط خودروهای زیادی تردد می کنند و زمان زیادی هم در پارکینگ هستند، برای سازماندهی بهتر پارکینگ ها نیازی برای مدیریت آن ها به صورت هوشمند و مفید بوجود آمده است.

پایگاه داده ی مورد نظر این نیاز را با ذخیره ی اطلاعات مربوط به هر پارکینگ، مشتری ها، و رزروها و پرداختی ها برآورده می کند.

این پایگاه داده 8 جدول دارد که به شرح زیر است:

**پارکینگ Parking:** این جدول وظیفه ی ذخیره کردن اطلاعات هر پارکینگ را برعهده دارد که شامل آیدی، نام، آدرس، وجود امکان رزرو جای پارک (اینکه آیا پارکینگ در کل خدمات رزرو را ارائه می دهد یا نه)، و تعداد طبقات است.

متغیر emptySlot حذف شد چون می توان بدون اینکه حافظه ای درگیر شود، آن را با query بدست آورد.

**طبقه Floors:** این جدول وظیفه ی ذخیره کردن اطلاعات هر طبقه ی پارکینگ را برعهده دارد که شامل آیدی، شماره طبقه، و تعداد کل جای پارک ها است.

متغیر emptySlot حذف شد چون می توان بدون اینکه حافظه ای درگیر شود، آن را با query بدست آورد.

**جای پارک Parking\_slot:** این جدول وظیفه ی ذخیره کردن اطلاعات هر جای پارک را برعهده دارد که شامل آیدی، آیدی طبقه، و وضعیت پر یا خالی بودن آن است.

**مشتری Customer:** این جدول وظیفه ی ذخیره کردن اطلاعات مشتریان را برعهده دارد که شامل آیدی، نام، شماره پلاک خودرو، تاریخ ثبت نام اولیه، و شماره تماس است.

**parking\_customer:** این جدول بدلیل اینکه ممکن است هر فرد در چند پارکینگ عضو شده باشد ایجاد شد که شامل آیدی، آیدی پارکینگ، و آیدی مشتری است.

**پرداخت Payment:** این جدول وظیفه ی ذخیره کردن اطلاعات پرداختی ها را برعهده دارد که شامل آیدی، تاریخ پرداخت، مقدار پرداخت، و نوع پرداخت (نقد یا آنلاین) است.

**رزرو parking\_slot\_reservation:** این جدول وظیفه ی ذخیره ی اطلاعات رزروهای جای پارک ها را برعهده دارد. شامل آیدی، آیدی مشتری، آیدی جای پارک، تاریخ شروع، تاریخ پایان، و مقدار کل مبلغی که باید پرداخته شود.

**reservation\_payment:** این جدول بدلیل اینکه ممکن است هر رزرو چند پرداختی داشته باشد ایجاد شده است که شامل آیدی، آیدی رزرو، و آیدی پرداخت است.

## Stored procedure:

### Procedure اول (parking\_customers)

اطلاعات همه ی مشتری های یک پارکینگ به خصوص را برمیگرداند.  
ورودی آن آیدی پارکینگ است و خروجی آن یک جدول از اطلاعات مشتری های آن پارکینگ است.

### Procedure دوم (average\_payment)

میانگین مبالغ دریافتی هر پارکینگ را برمیگرداند.  
ورودی ندارد و خروجی آن جدولی از آیدی و نام و میانگین مبالغ دریافتی هر پارکینگ موجود در سیستم است.  
این procedure بدلیل تغییرات در پایگاه داده نسبت به قبل تغییر کرده است.

### Procedure سوم (last\_customerID)

آیدی آخرین مشتری یک جای پارک به خصوص را برمیگرداند.  
ورودی آن آیدی جای پارک است و خروجی آن آیدی آخرین مشتری این جای پارک است.

## Functions:

### تابع اول (available\_parking\_slots)

این تابع جدولی از جای پارک های خالی در یک طبقه ی خاص پارکینگ در یک بازه زمانی خاص را برمیگرداند.  
ورودی آن آیدی پارکینگ، آیدی طبقه، تاریخ شروع و تاریخ پایان است و خروجی آن مشخصات جای پارک های خالی طبقه و پارکینگ ورودی در بازه زمانی ورودی است.  
به این صورت که بررسی میکند تاریخ شروع رزرو ها بعد از انتهای بازه ورودی باشد یا تاریخ پایان رزرو ها قبل از ابتدای بازه ورودی باشد.

### تابع دوم (parking\_name)

این تابع اطلاعات پارکینگ ها با اسمی مشخص را برمیگرداند.  
ورودی تابع رشته ای به عنوان نام پارکینگ است و خروجی آن اطلاعات همه ی پارکینگ ها با آن نام است.  
علاوه بر مشخصاتی که در table پارکینگ است، تعداد جای پارک های خالی هم نشان داده میشود.

### تابع سوم (parking\_TotalEmptySlots)

این تابع تعداد جاهای خالی پارکینگ به خصوصی را برمیگرداند.  
ورودی آن آیدی پارکینگ و خروجی آن تعداد جای پارک های خالی آن پارکینگ است.

### تابع چهارم (parking\_list)

این تابع جدولی از مشخصات پارکینگ هایی که یک فرد به خصوصی مشتری آن هاست را برمیگرداند.  
ورودی آن آیدی مشتری است و خروجی آن جدولی از مشخصات پارکینگ هایی است که آن فرد، مشتری آن هاست. علاوه بر مشخصاتی که در table پارکینگ است، تعداد جای پارک های خالی هم نشان داده میشود.

## Views:

### View اول (view\_reservation)

اطلاعات رزروهایی را برمیگرداند که هنوز پرداختشان کامل نشده است.

### View دوم (parking\_total\_payment)

مجموع همه ی پرداختی های هر پارکینگ را برمیگرداند.

دو view اول بدلیل تغییر در پایگاه داده، تغییراتی داشته اند.

### View سوم (empty\_slots\_num)

تعداد جای پارک های خالی را به تفکیک هر طبقه و هر پارکینگ میدهد. بدین منظور از rollup استفاده شده است.

## Triggers:

### تریگر اول (paymentamount\_reservation)

این تریگر بعد از اضافه شدن یک ردیف به parking\_slot\_reservation فعال میشود و مقدار مبلغ آن رزرو را طبق تعداد روزها و هزینه ی جای پارکی که مشتری قصد رزرو دارد محاسبه میکند و برای فیلد paymentamount قرار میدهد. اینگونه از اشتباهات دستی جلوگیری میشود.

### تریگر دوم (check\_amount)

این تریگر پس از اضافه شدن یک ردیف به جدول payment فعال میشود. بررسی میکند که اگر مجموع پرداختی ها از مقدار مورد نیاز برای پرداخت paymentamount در جدول parking\_slot\_reservation بیشتر بود، یک پیام آلارم چاپ میشود و این موضوع را هشدار می دهد.

دو تریگر اول بدلیل تغییر در پایگاه داده، تغییراتی داشته اند.

تریگر های (payment\_insert)، (payment\_delete) و (payment\_update) تغییرات جدول payment شامل update, delete, insert را در جدولی به نام PaymentLogs ذخیره میکند.