

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی صنایع

گزارش پروژه درس سیستم های اطلاعات مدیریت

نگارش

روشا مشتاقیان ۹۸۲۵۰۸۰

حامد اعراب ۹۹۲۵۰۰۳

استاد درس

استاد حاجی زاده

استاد راهنما

استاد علی روشن ضمیران

فهرست مطالب

۳.....	مقدمه
۳.....	اهمیت طراحی پایگاه داده برای سرویس تاکسی آنلاین

تعاریف نمودار ER

۴.....	موجودیت
۴.....	ویژگی
۵.....	ارتباطات

جدول

۶.....	نحوه ترسیم
۷.....	نحوه انجام پروژه و نتایج
۱۲.....	استفاده از power bi

مقدمه

این گزارش درباره اهمیت طراحی پایگاه داده در موضوع انتخابی توضیح میدهد همچنین گام به گام نحوه کشیدن نمودار ER و تبدیل آن به جدول و در آخر کد نویسی در SQL Server را شرح میدهد. نمودارها و خروجی ها در برنامه های استفاده شده به صورت عکس در این گزارش ارائه شده است.

موضوع انتخابی گروه سرویس تاکسی آنلاین است که به صورت گسترده در آن داده ردوبدل میشود و منظم سازی و استفاده از آنها ضروری است.

اهمیت طراحی پایگاه داده برای سرویس تاکسی آنلاین

فناوری در تاکسیرانی، موضوعی جدید است که هر سال کشورهای مختلفی سعی می کنند با ارائه فناوری های جدید، زمینه ای برای راحت تر کردن استفاده از سیستم های حمل و نقل رو محیا کنند. یکی از مهم ترین دلایل پیشرفت تو حوزه ی فناوری در تاکسیرانی و مسافربری استفاده از اپلیکیشن های مربوط به این زمینه هست که می توان با دانلود و مراجعه به این برنامه های کاربردی در هر ساعت از شبانه روز، سرویس مورد نظر را درخواست کرد. این اپلیکیشن ها با ساختاری منظم و کاربر پسند، محیطی را برای کاربر ارائه میدهند که علاوه بر درخواست خودرو برای جابه جایی بتوان مبدا و مقصد رو به راحتی برای راننده معین کرد و به صورت آنلاین هزینه سفر را برای راننده پرداخت کرد. وجود پایگاه داده در سرویس تاکسی آنلاین، همه اطلاعات را در یک محیط خاص جمع میکند که اجازه ی ذخیره، پردازش و بازیابی اطلاعات را میدهد، همچنین کاربران میتوانند به راحتی به داده ها دسترسی داشته باشند تا بتوان مشکلات را برطرف کرد و برای توسعه سرویس تلاش کرد. همچنین باید توجه کرد که سرویس تاکسی آنلاین موظف است امنیت مسافران را تضمین کند که وجود این اطلاعات سازمان یافته این کار را تسهیل میکند.

تعاریف نمودار ER

موجودیت

پدیده یا هر شی که در آن داده ذخیره می شود و راه تشخیص آن امکان تقسیم به اجزای کوچکتر است که باید حداقل یکی از شرط های زیر را داشته باشد:

- ۱- حداقل یک ویژگی غیرکلیدی داشته باشد.
 - ۲- در رابطه چند به یک، موجودیت سر چند باشد.
- با توجه به توضیح موجودیت، گزینه های زیر برای موجودیت سرویس تاکسی آنلاین انتخاب شد:

- راننده
- سفر
- مسافر
- پشتیبانی
- رسید حساب راننده یا مسافر
- درخواست پشتیبانی از طرف راننده یا مسافر

ویژگی

هر موجودیت ویژگی هایی مربوط به خودش را دارد، به هر یک از این ویژگی ها صفت گفته میشود همچنین هر موجودیت باید ویژگی یا ویژگی های منحصر به فردی داشته باشد که آن را از سایر موجودیت ها متمایز کند، بنابراین صفت کلیدی تعریف می شود که برای نمایش زیر آن صفات خط می کشیم.

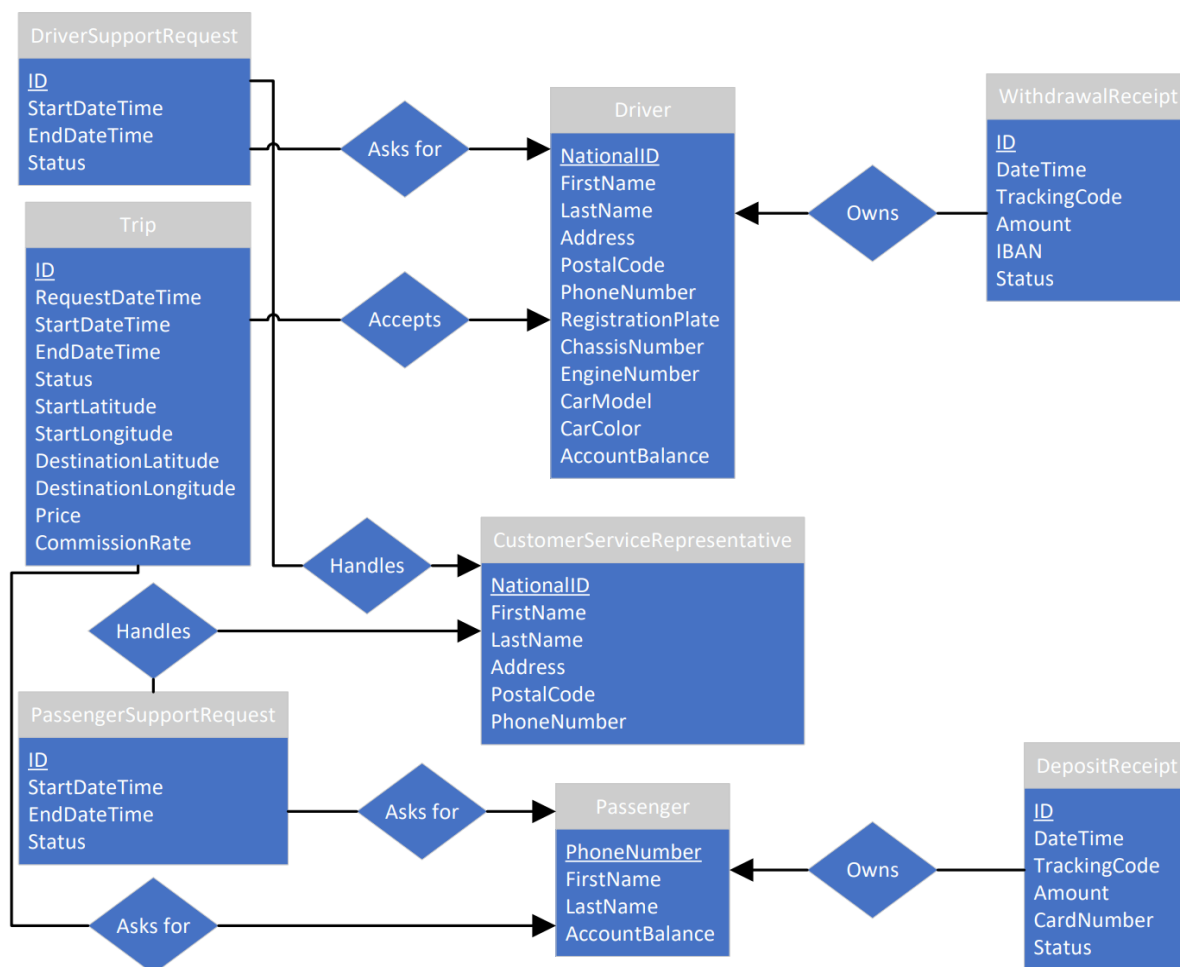
- راننده = کد ملی راننده - نام - نام خانوادگی - آدرس - شماره تلفن - کد پستی - مدل ماشین - رنگ ماشین - شماره موتور - شماره شاسی - موجودی حساب
- سفر = کد سفر - تاریخ و ساعت درخواست و شروع و پایان - مبدا - مقصد - وضعیت - مبلغ - مدت زمان
- مسافر = شماره تلفن - نام - نام خانوادگی - موجودی حساب
- پشتیبانی = کد ملی پشتیبان - نام - نام خانوادگی - آدرس - شماره تلفن - کد پستی

- رسید حسابها= شناسه- موجودی- وضعیت- کد رهگیری- شماره کارت- تاریخ
- درخواست ها= شناسه- تاریخ و ساعت شروع و پایان درخواست- وضعیت

ارتباطات

ارتباط های استفاده شده در این پروژه، یک به چند یا **one-many** است. به این معنا که چند نمونه از موجودیت یکی حداکثر با یک نمونه از موجودیت دیگری ارتباط دارد ولی همین ارتباط برعکس با چند نمونه ارتباط دارد.

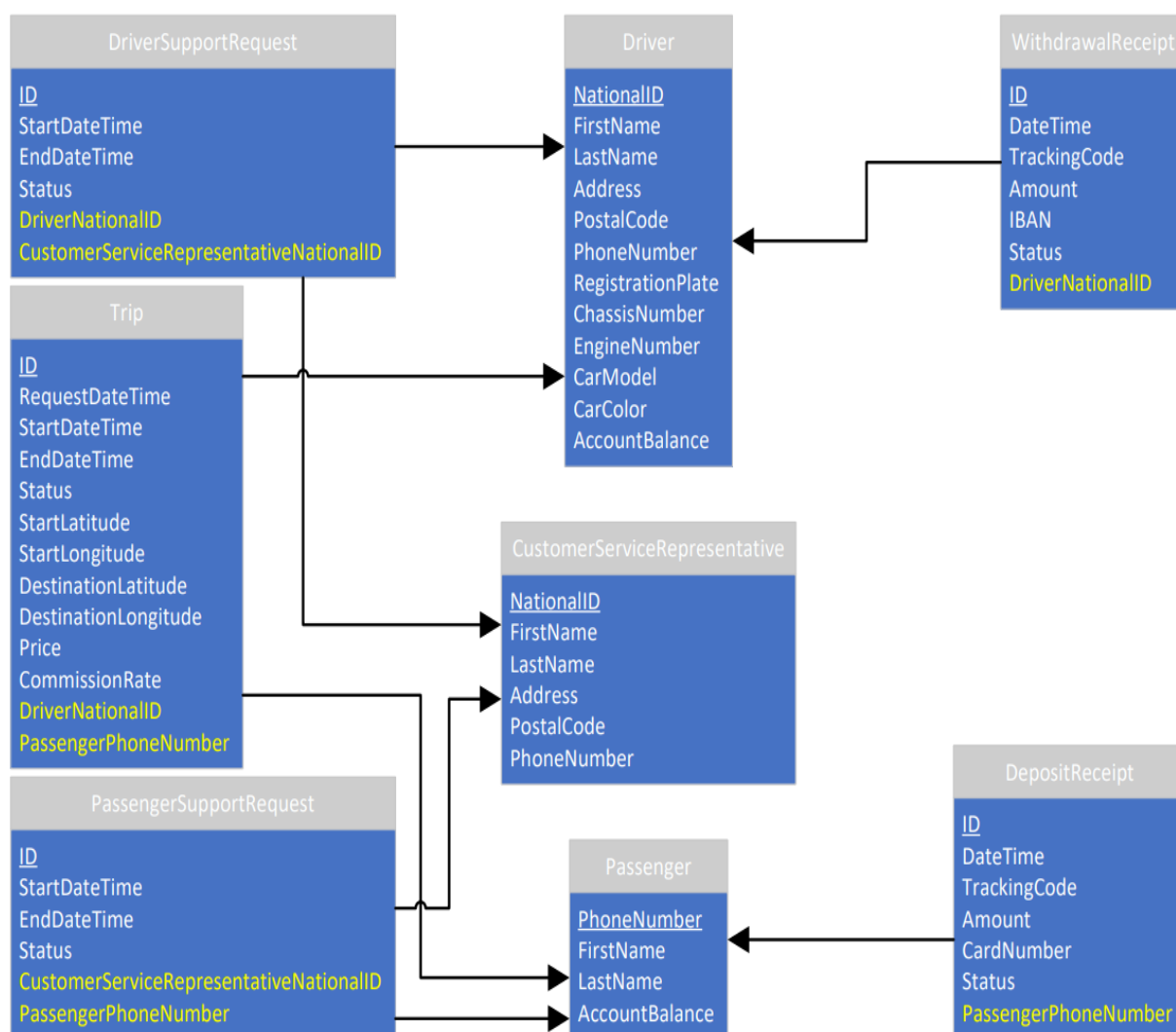
برای نشان دادن ارتباط بین دو موجودیت از فلش های ارتباطی و لوزی به عنوان شرط استفاده می شود.



جدول

نحوه ترسیم

نحوه ترسیم روابط جدول هم مانند نمودار ER است اما تمامی موجودیت ها تبدیل به جدول میشوند همچنین با توجه به اینکه تمامی روابط یک به چند هستند پس لوزی ها حذف میشوند. ویژگی های کلیدی سر چند فلش در جدول موجودیت که سر یک فلش است قرار میگیرد و به عنوان کلید خارجی در خود ثبت میکند.



نحوه انجام پروژه و نتایج

در ابتدا با توجه به تعاریف، نمونه سیستم رسم شد و مشکلات آن روی کاغذ اصلاح شد سپس نمودار ER آن در Visio رسم شد سپس آن را در SQL تبدیل به جدول کردیم. داده های رندوم و انتخابی وارد هر ویژگی کردیم و در آخر نتایج مورد نظر را از داده ها بدست آوردیم و برای بهبود سیستم راهکار پیشنهاد دادیم.

۱. طبق داده ها، کد پشتیبانی دهنده ای که بیشترین درخواست را پاسخگو بود را یافتیم و برای او پاداشی در نظر میگیریم.

```
SELECT TOP 1 CustomerServiceRepresentativeNationalID, COUNT(CustomerServiceRepresentativeNationalID) AS SupportsCount
FROM DriverSupportRequest
GROUP BY CustomerServiceRepresentativeNationalID
```

	CustomerServiceRepresentativeNationalID	SupportsCount
1	6908363272	19

۲. پشتیبانی دهنده ای که کمترین درخواست را انجام داده است را یافتیم تا برای او انگیزه ای ایجاد کنیم و در آینده اگر کم کاری رفع نشد او را از سازمان برکنار کنیم.

```
SELECT TOP 1 CustomerServiceRepresentativeNationalID, COUNT(CustomerServiceRepresentativeNationalID) AS SupportsCount
FROM DriverSupportRequest
GROUP BY CustomerServiceRepresentativeNationalID
ORDER BY SupportsCount ASC
```

	CustomerServiceRepresentativeNationalID	SupportsCount
1	8427168270	8

۳. از بین مسافران و تعداد سفر آنها، ۱۰ مسافری که کمترین سفر را درخواست کرده را پیدا کرده و تخفیف هایی برایشان در نظر گرفتیم تا بیشتر از این سیستم استفاده کنند.

```
SELECT TOP 10 PassengerPhoneNumber, COUNT(PassengerPhoneNumber) AS TripsCount
FROM Trip
GROUP BY PassengerPhoneNumber
ORDER BY TripsCount ASC
```

	PassengerPhoneNumber	TripsCount
1	9357319658	2
2	9818351033	2
3	9449778896	3
4	9760481200	3
5	9516268229	3
6	9697541581	3
7	9871824453	4
8	9443534635	4
9	9628061763	4
10	9744285827	4

۴. همچنین ۱۰ مسافری که بیشترین فعالیت را داشته و فعال بودند را یافتیم و برای آنها امتیازات و جوایزی در نظر میگیریم.

```
SELECT TOP 10 PassengerPhoneNumber, COUNT(PassengerPhoneNumber) AS TripsCount
FROM Trip
GROUP BY PassengerPhoneNumber
ORDER BY TripsCount DESC
```

	PassengerPhoneNumber	TripsCount
1	9565800940	19
2	9205418572	18
3	9006707393	18
4	9623244587	18
5	9057407351	17
6	9318867479	17
7	9175590208	17
8	9420879893	17
9	9143775492	17
10	9223236861	17

۵. میانگین زمانی که برای پاسخگویی به درخواست های پشتیبانی و برطرف کردن آن صرف می شود را بدست آوردیم تا در آینده در تلاش برای کاهش این زمان باشیم. (در ابتدا برای درخواست پشتیبانی از طرف راننده را یافتیم سپس از طرف مسافر یافت شد).

```
SELECT AVG(DATEDIFF(minute, StartDateTime, EndDateTime)) as AverageDriverSupportRequestDurationInMinutes
FROM DriverSupportRequest
```

	AverageDriverSupportRequestDurationInMinutes
1	700

```
SELECT AVG(DATEDIFF(minute, StartDateTime, EndDateTime)) as AveragePassengerSupportRequestDurationInMinutes
FROM PassengerSupportRequest
```

	AveragePassengerSupportRequestDurationInMinutes
1	695

۶. میانگین زمان قبول شدن سفر توسط راننده ها بدست آمد تا این زمان را کاهش دهیم و درخواست مسافرها با سرعت بیشتری پذیرفته شود.

```
SELECT AVG(DATEDIFF(minute, RequestDateTime, StartDateTime)) as AverageDriversResponseTimeInMinutes
FROM Trip
```

	AverageDriversResponseTimeInMinutes
1	3

۷. از داده ها پنج راننده فعال استخراج شد و برای آنها ترفیع یا درآمد بیشتری در نظر گرفتیم.

```
SELECT TOP 5 DriverNationalID, COUNT(DriverNationalID) AS TripsCount
FROM Trip
GROUP BY DriverNationalID
ORDER BY TripsCount DESC
```

	DriverNationalID	TripsCount
1	8460174143	117
2	2801941245	117
3	1966186737	117
4	3954404377	114
5	1204695280	114

۸. همچنین پنج راننده کم فعال هم یافتیم تا برای آنها انگیزه ایجاد کنیم.

```
SELECT TOP 5 DriverNationalID, COUNT(DriverNationalID) AS TripsCount
FROM Trip
GROUP BY DriverNationalID
ORDER BY TripsCount ASC
```

	DriverNationalID	TripsCount
1	8462207009	72
2	2745133927	79
3	7927263732	84
4	8552695650	85
5	8708259550	86

۹. درصد وضعیت سفرها را پیدا کرده تا میزان کنسلی را پیدا کنیم و برای کاهش میزان کنسلی سفرها تلاش کنیم و با درصد های آینده مقایسه کنیم.

```
SELECT Status, COUNT(*) * 100 / SUM(COUNT(*)) OVER() AS Percentage
From Trip
GROUP BY Status
```

	Status	Percentage
1	Canceled	8
2	Done	91

۱۰. درصد وضعیت های رسیدهای برداشت و واریز به ترتیب بدست آمد تا درصد ناموفق بودن را بهبود ببخشیم و برای ارزیابی با درصد های آینده مورد استفاده قرار بگیرند.

```
SELECT Status, COUNT(*) * 100 / SUM(COUNT(*)) OVER() AS Percentage
From WithdrawalReceipt
GROUP BY Status
```

	Status	Percentage
1	Failed	9
2	Successful	91

```
SELECT Status, COUNT(*) * 100 / SUM(COUNT(*)) OVER() AS Percentage
From DepositReceipt
GROUP BY Status
```

	Status	Percentage
1	Successful	100

۱۱. میانگین مدت زمان سفرها را بدست آوردیم تا میزان طولانی بودن سفرها را بررسی کنیم.

```
SELECT AVG(DATEDIFF(minute, StartDateTime, EndDateTime)) as AverageTripDurationInMinutes
FROM Trip
```

	AverageTripDurationInMinutes
1	29

۱۲. درصد وضعیت درخواست های پشتیبانی از طرف راننده و مسافر به ترتیب بدست آمد تا میزان درخواست های حل شده را با در حال انجام گیری، مقایسه کنیم و سرعت پاسخگویی را بالا ببریم.

```
SELECT Status, COUNT(*) * 100 / SUM(COUNT(*)) OVER() AS Percentage
From DriverSupportRequest
GROUP BY Status
```

	Status	Percentage
1	Processing	0
2	Resolved	99

```
SELECT Status, COUNT(*) * 100 / SUM(COUNT(*)) OVER() AS Percentage
From PassengerSupportRequest
GROUP BY Status
```

	Status	Percentage
1	Resolved	99
2	Processing	0
3	WaitingF...	0

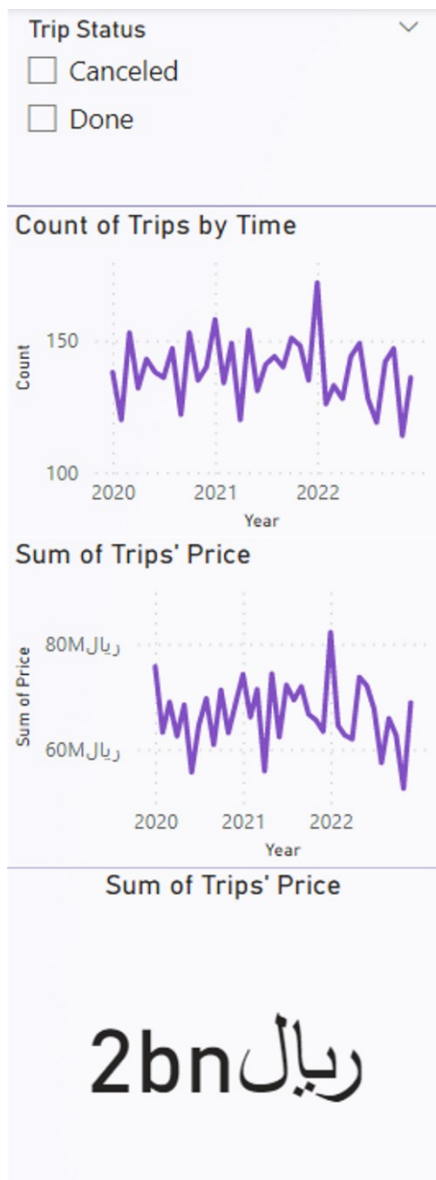
۱۳. طبق ساعات ثبت شده، تعداد سفرها در ساعات شبانه روز یافت شد تا ساعاتی که بیشترین درخواست سفر رخ میدهد را پیدا کنیم و سرویس بهتری ارائه دهیم.

```
SELECT DATEPART(hour, RequestDateTime) AS RequestHour, COUNT(DATEPART(hour, RequestDateTime)) AS Total
FROM Trip
GROUP BY DATEPART(hour, RequestDateTime)
ORDER BY Total DESC
```

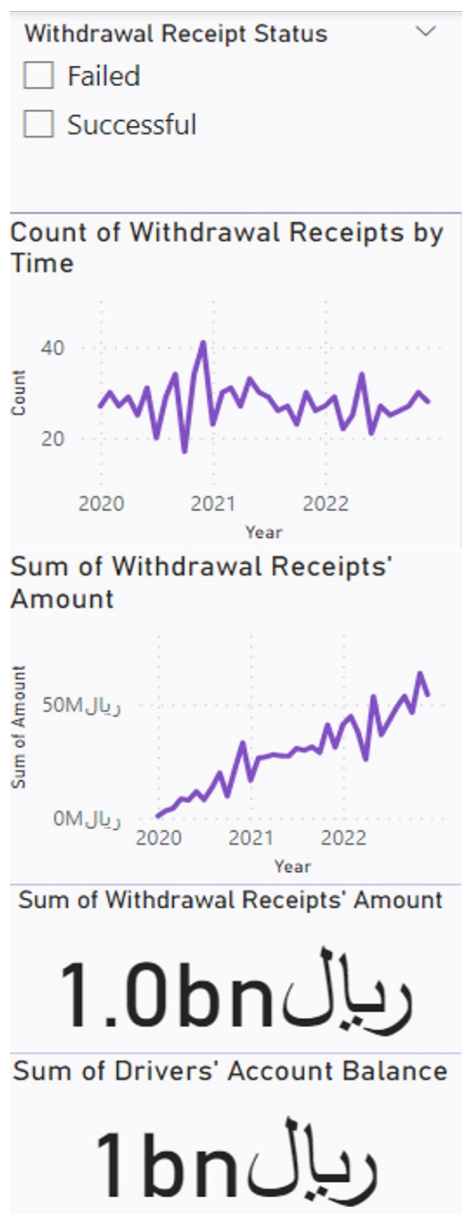
	RequestHour	Total
1	18	230
2	17	224
3	11	224
4	0	221
5	8	219
6	6	215
7	10	215
8	16	215
9	14	213
10	20	211
11	2	211
12	4	209
13	22	209
14	7	207
15	5	206
16	21	205
17	15	201
18	3	200
19	12	200
20	1	198
21	13	197
22	23	196
23	9	191
24	19	183

استفاده از Power BI

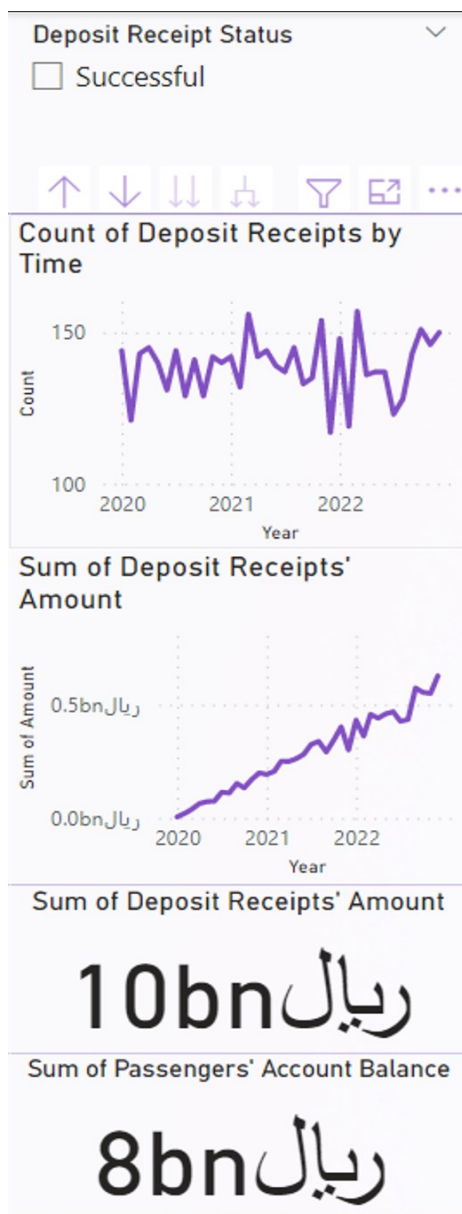
Measure که در فارسی به سنجه آماری یا معیار ترجمه میشود در اصل یک فرمول محاسباتی هست که بر مبنای Filter Context یک خروجی عددی، متنی یا تاریخی برامون تولید میکنه. برای ساخت یک سنجه، در Report View، از تب مدلینگ New Measure را انتخاب کردیم. یکی از خوبی های زبان عبارات آنالیز داده ها در Power BI، این است که توابع مفید زیادی بخصوص پیرامون محاسبات متکی بر زمان مانند سال و تاریخ و یا مقایسه داده های سال به سال دارد که ما از داده های خود چند نمودار متفاوت جهت تحلیل بهتر تولید کردیم که در زیر توضیح داده شده است.



- طبق مجموعه نمودارهای زیر که در رابطه با سفرهای انجام شده در گذشته است، در ابتدا تعداد سفرهای درخواست شده در طول سالها به نمایش گذاشته شد و همانطور که قابل مشاهده است اوج سفرهای انجام شده در اواخر سال ۲۰۲۱ است. در نمودار دوم هزینه سفرهای درخواست شده در هر ماه جمع شده است تا خلاصه ای از میزان استفاده مسافرها داشته باشیم و در آخر جمع هزینه های کل سفرها از ابتدا شروع کار تا به امروز داده شده است تا ارزش کل خدمت ارائه شده را بدانیم.



- در نمودار اول شاهد روند تعداد تراکنش راننده های این سیستم در هر ماه هستیم همچنین در نمودار دوم به مجموع این تراکنش ها، ماهیانه پرداخته ایم که نشان می دهد راننده ها به طور متوسط هر ماه چه مبلغی از حساب خود برداشت میکنند و در آخر مجموع کل برداشت و کل موجودی حساب آنها از شروع سیستم آمده است.



- نمودار خطی اول نشان دهنده تعداد واریز های انجام شده توسط مسافر در هر ماه است که میزان فعال بودن مسافران را به نمایش میگذارد. در نمودار بعدی جمع مبلغ واریز ها به صورت ماهیانه بدست آمده و در آخر جمع کل واریزی ها از طرف مسافر و جمع موجودی حساب آنها نشان داده شده است.

- نتایج بدست آمده بالا در ارتباط با درخواست های پشتیبانی از طرف راننده و مسافر است که در سمت راست تعداد درخواست های راننده به صورت ماهیانه و در سمت چپ مربوط به مسافر است. همچنین تعداد راننده ها به تفکیک مدل ماشین آنها و منطقه سکونت آنها نیز مشخص است و در آخر تعداد کل مسافرها و راننده ها محاسبه شده است.

