

دوره کامل و رایگان

Power Query UI

مدرس: پوریا بغدادی
گردآورنده: فاطمه جواهری

❖ تعداد جلسات: ۱۰

❖ مدت آموزش: ۱۰ ساعت

❖ تعداد صفحات: ۱۳۱

فهرست

جلسه اول	۱
هدف نرم افزار Power Query چیست؟	۱
حل مشکلات قدیمی اکسل با روش های مدرن و آسان	۱
خودکارسازی فرآیندهای آماده سازی داده، گزارش گیری و مغایرت گیری	۱
ادغام و آماده کردن فایل های مختلف	۱
کاهش وابستگی به برنامه نویسی	۱
مخاطبان Power Query چه کسانی هستند؟	۲
منبع آموزشی	۲
چیستی و قابلیت ها	۲
واژه ها و اصطلاحات	۳
نحوه دسترسی و استفاده	۳
محیط و امکانات	۵
منابع آموزشی Power Query	۸
ویژگیهای اصلی بخش Data Preview	۹
Lazy Load	۹
Cells and errors	۹
مدیریت خطاها و ابزارهای کاربردی در Power Query	۱۱
Column & Row ..	۱۲
Go to Column	۱۳
Choose Columns	۱۳
Formula Bar	۱۴
Basic Column Actions	۱۴
Detect Data Type	۱۴
Column Profile	۱۵
Column Quality	۱۶
Column Distribution	۱۶
Reorder Columns	۱۷
Sort & Filter	۱۷
Remove / Remove Other Columns	۱۸
Duplicate Column	۱۸

۱۹	Rename Columns
۱۹	Move Columns
۱۹	Replace Value
۲۰	Copy
۲۱	جلسه دوم
۲۱	دستورات پیشرفته ستون ها
۲۱	Fill Up ,Fill Down
۲۳	Remove Duplicate
۲۶	Split to Column
۲۸	Remove Error/ Replace Error
۳۲	Rename Column
۳۳	Drill Down
۳۴	Transform
۳۵	Move Column
۳۵	Add As New Query
۳۷	جلسه سوم
۳۷	Transform
۳۷	Group By
۳۸	Use First Row as Headers
۴۰	Transpose
۴۱	Reverse Rows
۴۱	Count Rows
۴۲	Data Type
۴۲	Detect Data Type
۴۲	Rename
۴۳	Replace Value
۴۳	Fill Down / Fill Up
۴۳	Pivot Column
۴۴	Unpivot Columns
۴۵	Move
۴۵	Convert to List
۴۶	Split Column

۴۶		Format
۴۷		Extract
۴۷		Parse
۴۸		Statistics
۴۹		Standard
۴۹		Scientific
۴۹		Trigonometry
۵۰		Rounding
۵۰		Information
۵۰		Date & Time
۵۱		Create Data Type
۵۲		جلسه چهارم .
۵۲		Merge & Append
۵۲		Append دستور
۵۷		Merge دستور
۵۹		انجام عملیات Merge
۶۲		انواع (Join Types) Merge در Power Query
۷۰		جلسه پنجم.
۷۰		Import Folder
۷۹		جلسه ششم.
۷۹		Parameter و Reduce Rows
۷۹		Parameter
۸۵		تعریف مسیر فایل داینامیک
۸۶		Reduce Rows
۸۷		Remove Rows در Power BI / Power Query
۸۹		Remove Rows
۹۱		جلسه هفتم.
۹۱		Table، Reference و Duplicate
۹۱		Table
۹۳		Reference
۹۸		جلسه هشتم.
۹۸		ایجاد تقویم شمسی و جدول Dimedate

۹۹	نحوه ساخت Date (Dim Date)
۱۰۱	محاسبه فاصله بین تاریخ‌ها
۱۰۴	جلسه نهم
۱۰۴	مثالهای کاربردی
۱۰۴	مسئله اول -
۱۰۵	مسئله ۲-
۱۰۷	مسئله ۳ - اصلاح ساختار جدول (Unpivot Columns) در Power Query
۱۰۹	مسئله ۴ - رفع ادغام سلول‌ها و تبدیل داده‌های سطری به ستونی در Power Query
۱۱۰	مسئله ۵
۱۱۲	مسئله ۶ - تبدیل جدول شامل نام شرکت‌ها و ماه‌ها به ساختار ستونی استاندارد در Power Query
۱۱۵	مسئله ۷ - جدول با دو هدر (دو ردیف عنوان)
۱۱۷	مسئله ۸ - ترکیب سطرها و ستون‌ها برای ساخت جدول تاریخ و مقدار
۱۱۹	جلسه دهم
۱۱۹	زبان M
۱۲۲	مشاهده و شناسایی توابع در (Power Query (M Language)
۱۲۴	منابع یادگیری زبان M
۱۲۵	دلایل استفاده از زبان M
۱۲۵	دلیل اول: کند شدن فایل‌های سنگین
۱۲۷	دلیل دوم: پاک‌سازی سرستون‌های یک (Header) کثیف در Power Query
۱۲۸	دلیل سوم: فیلتر بدون حساسیت به حروف بزرگ/کوچک
۱۲۹	دلیل چهارم: خودکارسازی فرایندهای پیچیده
۱۲۹	دلیل پنجم: ایجاد جدول سفارشی (Custom Table)
۱۲۹	دلیل ششم: تغییر و تبدیل نوع داده‌ها
۱۳۰	دلیل هفتم: افزودن دیتا در کدنویسی (Append Data / Insert Row)

جلسه اول

هدف نرم افزار Power Query چیست؟

حل مشکلات قدیمی اکسل با روش های مدرن و آسان

زیرا وقتی یک صفحه گسترده مثل اکسل داریم، دست ما را باز است و می توانیم داده ها را هرطور بخواهیم وارد کنیم؛ (نه مثل Database ها که ساختار دارند). اما گزارش گیری از این فایل فاجعه است و مجبوریم VBA و ماکرو بنویسیم. داده ها به صورت غیر ساختارمند هستند و این باعث می شود نتوان آن ها را مصورسازی کرد.

این طراحی و فرم ذخیره سازی در سیستم های تحلیل داده مثل Python، R و Power BI غیر قابل قبول است و انجام نمی شود. پس باید اصولی را در طراحی رعایت کنیم و آن جدول اولیه را به شکلی که می خواهیم تحلیل انجام شود تغییر دهیم. شاید بتوان بطور دستی اطلاعات را در اکسل ویرایش و به فرم استاندارد درآورد، اما اگر تعداد شرکت ها، محصولات یا ویزیتورها زیاد باشد، دیگر امکان پذیر نیست.

اگر بتوانیم داده ها را از ابتدا اصولی و درست طراحی کنیم، حدود ۸۰ تا ۹۰ درصد کار تحلیل داده ها انجام شده و مشکلات حل می شوند. اما ما همیشه با تعداد زیادی فایل اکسل روبه رو هستیم که از قبل ساخته شده و به همین شکل در دسترس ما قرار دارند.

خودکارسازی فرآیندهای آماده سازی داده، گزارش گیری و مغایرت گیری

اگر به همان فایل اکسل اطلاعات جدیدی اضافه شود، باید این فرآیند خودکار باشد. پس هم می خواهیم مشکلات را حل کنیم، هم این حل را تکرار پذیر کنیم و قابلیت استفاده مجدد داشته باشد.

ادغام و آماده کردن فایل های مختلف

این ادغام هم می تواند خودکارسازی شود.

کاهش وابستگی به برنامه نویسی

دیگر نیاز به کدنویسی های پیچیده نیست.

Power Query دو بخش دارد:

- UI یا همان User Interface (رابط کاربری)
- زبان M

بخش رابط کاربری امکانات نرم‌افزاری را به صورت بصری ارائه می‌دهد. در اکسل از تب Data → Get & Transform می‌توان وارد محیط UI شد. در این بخش کارها با کلیک و ماوس انجام می‌شود و نیازی به فرمول‌نویسی نیست.

زبان M همان کدهایی است که پشت‌صحنه در Power Query نوشته می‌شوند.

Power Query داخل نرم‌افزار اکسل از سال ۲۰۱۶ به بعد وجود دارد و نیازی به نصب جداگانه ندارد. Microsoft ابتدا Power Query را در اکسل عرضه کرد و پس از بازخورد کاربران در سراسر دنیا، آن را در سال ۲۰۱۵ به صورت جداگانه در نرم‌افزار Power BI قرار داد.

بنابراین Power Query هم در اکسل و هم در Power BI وجود دارد و آموزش مربوط به هر دو مشابه است. یعنی هر قابلیت که در Power Query در اکسل است، در Power BI نیز وجود دارد.

Power Query هیچ پیش‌نیاز نرم‌افزاری ندارد. فقط کافی است بدانیم داده چیست و چرا مشکلی به وجود آمده است. در واقع درک جدول و تصویرسازی ذهنی داده‌ها برای شروع کافی است.

اینطور می‌توان گفت که Power Query خودش پیش‌نیاز بسیاری از ابزارهای دیگر در دنیای تحلیل داده است.

مخاطبان Power Query چه کسانی هستند؟

رشته‌ها و مشاغل داده‌محور:

- حسابداری، مالی، صنایع، مدیریت، فناوری اطلاعات، و ...
- برنامه‌ریزی، کنترل پروژه، کنترل و تضمین کیفیت، فروش، بازاریابی، زنجیره تامین، تحقیقات بازار، تحلیل داده، هوش تجاری، بانکداری و

منبع آموزشی

بخش Learn سایت مایکروسافت و تجربیات مدرس در کار با داده‌های متنوع سازمانی از سال ۱۳۹۷ می‌باشد.

چیستی و قابلیت‌ها

Power Query داده‌ها را ادغام (Merge)، تمیز (Clean)، تجمیع (Aggregate) و تغییر فرم (Transform) می‌دهد و در قالب نرم‌افزاری که در آن استفاده می‌شود، خروجی را به ما تحویل می‌دهد.

با بخش UI در Power Query حدود ۸۰٪ تا ۹۰٪ مشکلات داده‌ها حل می‌شود. کافی است یکبار برای آن مراحل را تعریف کنیم، از دفعات بعد خودش به صورت خودکار اجرا می‌شود. ۱۰٪، ۲۰٪ بخش باقی‌مانده مربوط به زبان M است که برای انجام کارهای خاص‌تر بکار می‌رود.

واژه‌ها و اصطلاحات

Data Preparation

به معنای آماده‌سازی داده‌ها می‌باشد مانند اضافه کردن سطر، فیلتر کردن داده‌ها و انجام کارهای مقدماتی از این نوع.

Data Transformation

به معنای تغییر شکل داده‌ها می‌باشد.

Data Cleaning

به معنای پاک‌سازی داده‌ها می‌باشد، برای مثال، اگر داده‌ای اشتباه وارد شده باشد می‌توانیم آن را Replace کنیم.

اگر Space اضافی دارد می‌توانیم آن را حذف کنیم. اگر مجموعه‌ای از اطلاعات در یک ستون ادغام شده و می‌خواهیم آن را جدا کنیم تا چند ستون داشته باشیم، از قابلیت Split استفاده می‌کنیم.

ETL = Extract, Transform, Load

در این فرآیند:

- داده‌ها را با Power Query از منبعشان استخراج (Extract) می‌کنیم.
- تغییرات لازم را روی آن‌ها اعمال (Transform) می‌کنیم.
- و سپس در مقصد جدید بارگذاری (Load) می‌نماییم.

برای درک بهتر، می‌توان آن را مانند جریان آب پشت سد تصور کرد: آب اولیه شکلی خاص دارد (مثل داده خام). آن را از پشت سد می‌گیریم، آلودگی‌هایش را پاک می‌کنیم، ممکن است مواد خاصی مثل کلر به آن اضافه کنیم، و سپس در ظرفی جدید می‌ریزیم. این ظرف جدید همان داده‌ی پاک و ساختارمند است که از طریق لوله‌کشی یعنی فرآیند (ETL) منتقل می‌شود. Power Query یکی از معروف‌ترین ابزارهای ETL است. ابزارهای دیگری هم در این حوزه وجود دارند، مانند SSIS در نرم‌افزار Oracle ODI و SQL Server در نرم‌افزار Oracle.

نحوه دسترسی و استفاده

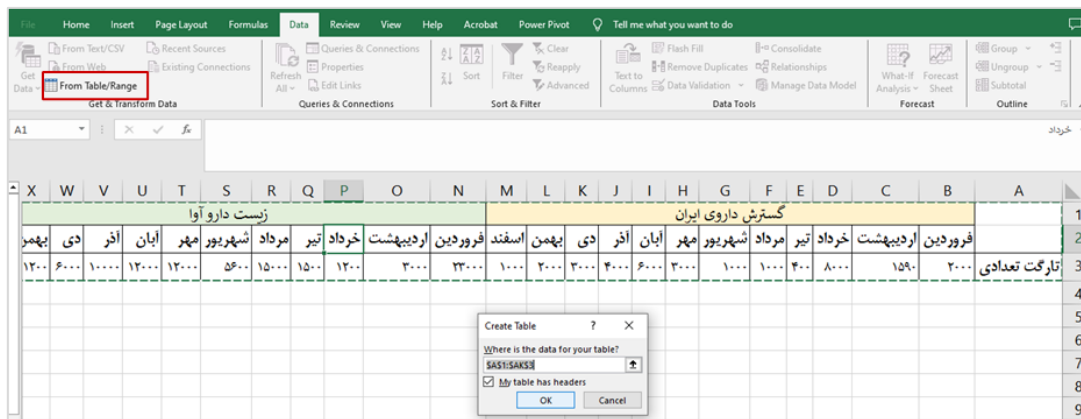
در اکسل از تب Data → Get Data می‌کنیم تا داده‌ی جدید وارد کنیم.

در Power BI هم از تب Home → Transform Data محیط Power Query باز می‌شود.

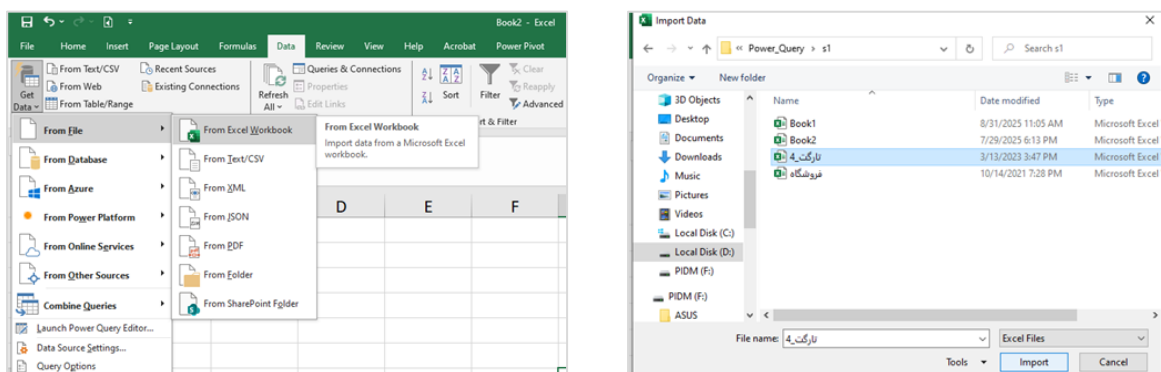
یادگیری Power Query در اکسل و Power BI هیچ تفاوتی ندارد، محیط و عملکرد در هر دو نرم‌افزار تقریباً

یکسان است.

وقتی در محیط اکسل هستیم و می‌خواهیم داده‌ها را به Power Query منتقل کنیم، کافی است روی محدوده‌ی داده کلیک کرده و از تب Data گزینه‌ی From Table/Range را بزنیم. این کار محدوده را به جدول تبدیل می‌کند و هم‌زمان داده را به محیط Power Query می‌برد.

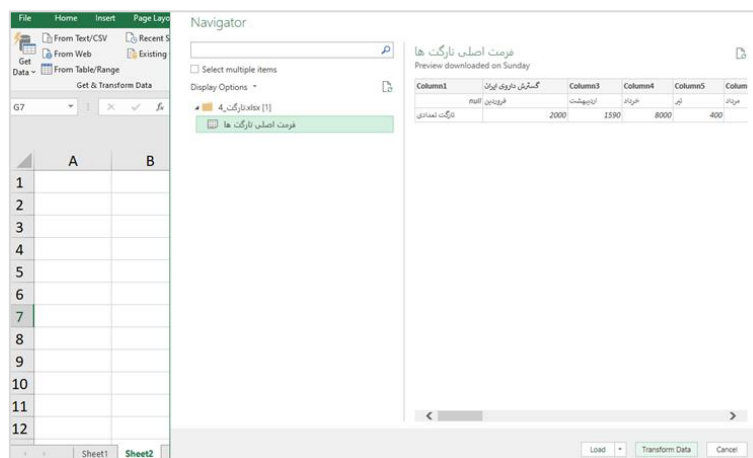


همچنین می‌توان یک فایل اکسل خالی باز کرد، از تب Data گزینه‌ی Get Data را انتخاب کرد، فایل موردنظر را Import نمود و سپس آن را Load یا Transform کرد تا در Power Query باز شود.

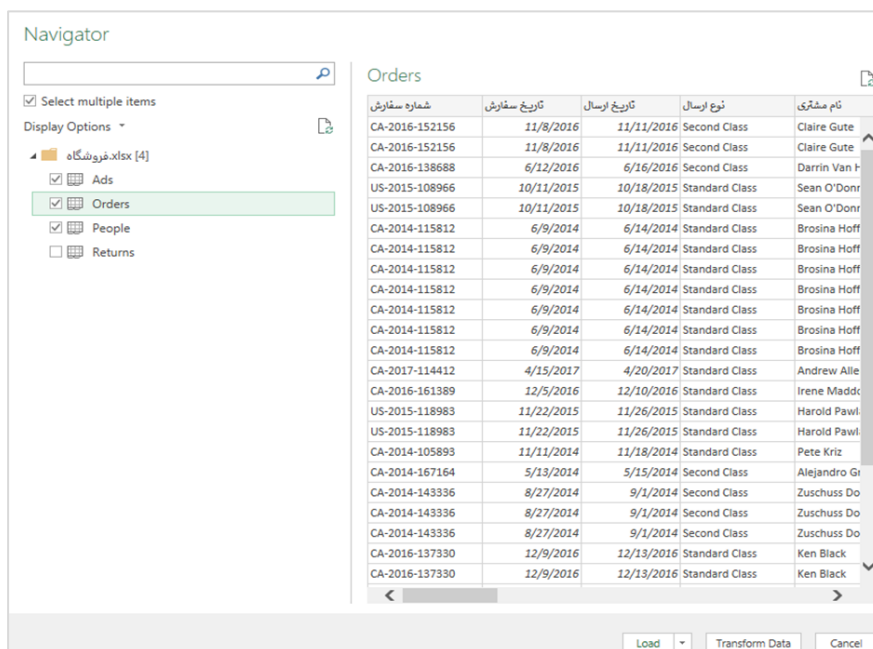


در بخش Navigator می‌توان تمام بخش‌های مختلف فایل اکسل را مشاهده کرد: جداول، شیت‌ها و سایر اجزای

موجود.

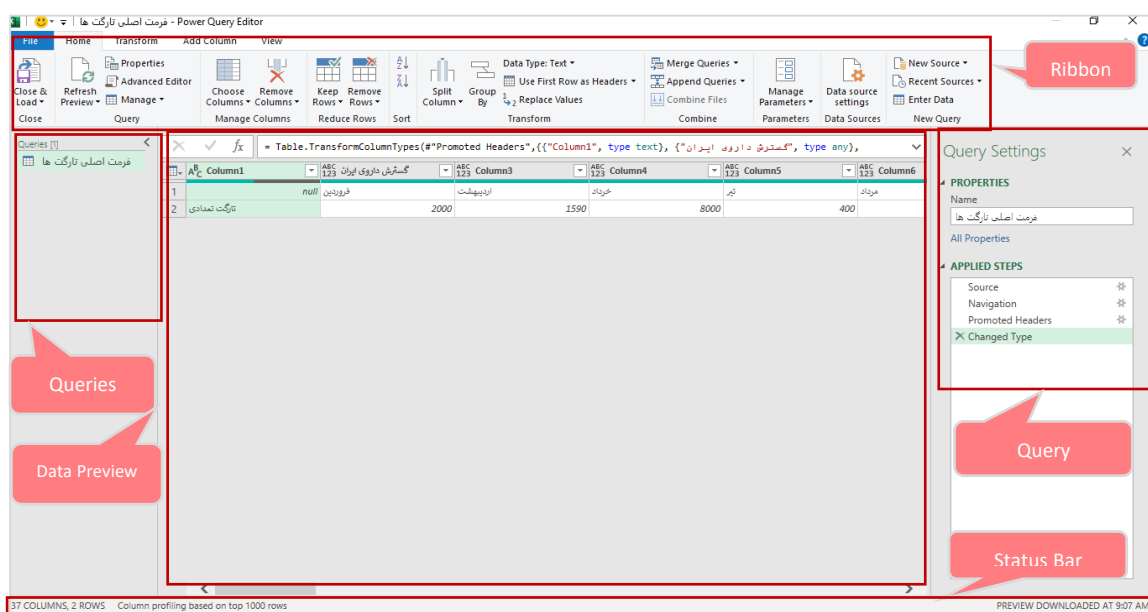


در Power BI نیز در بخش Navigator کنار نام هر شیت، یک Checkmark وجود دارد که باید آن را فعال کرد. اگر چند شیت مختلف وجود داشته باشد، می توان از گزینهی Select Multiple Items برای انتخاب همزمان آنها استفاده کرد.



محیط و امکانات

در محیط Power Query چند بخش اصلی وجود دارد که شناخت آنها برای کار با این ابزار ضروری است.



Ribbon

قسمت بالای صفحه شامل تب‌های مختلف است که به آن Ribbon گفته می‌شود. این بخش شامل تب‌هایی مانند Home، Transform، Add Column و View است. گرچه از نظر ظاهری شبیه تب‌های نرم‌افزار اکسل است، اما در واقع بخشی از محیط Power Query Editor محسوب می‌شود و در پشت‌صحنه با محیط اصلی اکسل تفاوت دارد.

Queries Pane

در سمت چپ صفحه، بخشی به نام Queries Pane قرار دارد. در این بخش، تمام Query هایی که ایجاد کرده‌ایم، نمایش داده می‌شوند. هر Query در واقع نشان‌دهنده‌ی یک منبع داده است.

برای مثال:

- اگر داده را از یک شیت اکسل خوانده باشیم، یک Query ایجاد می‌شود.
 - اگر داده را از SQL Server یا Access بیاوریم، باز هم یک Query ساخته می‌شود.
 - حتی اگر یک Function در Power Query تعریف کنیم، آن هم در اصل یک Query است.
- هر Query، پشت‌صحنه دارای کدهایی به زبان M است که از طریق گزینه‌ی Advanced Editor قابل مشاهده و ویرایش‌اند.

Data Preview

بخش میانی صفحه، Data Preview یا Current Preview نام دارد. در این قسمت محتوای داده‌ی هر Query نمایش داده می‌شود. یعنی با انتخاب هر Query در پنل سمت چپ، پیش‌نمایش آن در وسط صفحه نشان داده می‌شود.

Query Settings

در سمت راست صفحه بخشی به نام Query Settings وجود دارد که دو بخش اصلی دارد:

- **Properties** : شامل نام (Name) و توضیحات (Description) مربوط به کوئری است. می‌توان از اینجا نام Query را تغییر داد. (همچنین با Right Click → Rename روی Query در پنل سمت چپ هم قابل تغییر است).
- **Applied Steps** یکی از مهم‌ترین بخش‌های Power Query است. در این قسمت، تمام مراحل که روی داده انجام داده‌ایم (مثل فیلتر، ادغام، حذف ستون و ...) به ترتیب ثبت می‌شود. با هر بار Refresh که داده‌ی جدید وارد شود، تمام این مراحل به‌صورت خودکار دوباره روی داده‌های جدید اعمال می‌شوند.

به همین دلیل Power Query یک محیط تکرارشونده و خودکار است. مراحل در Applied Steps قابل ویرایش، حذف، جابه‌جایی و حتی بازسازی هستند. همچنین گام‌هایی که کنارشان آیکون چرخ‌دنده دارند قابل تنظیم‌اند، و گام‌هایی که چرخ‌دنده ندارند، مستقیماً با زبان M نوشته شده‌اند و می‌توان از طریق Formula Bar یا Advanced Editor آن‌ها را ویرایش کرد.

در تب View می‌توان Formula Bar را فعال یا غیرفعال کرد. در این نوار، کد مربوط به هر مرحله‌ی انتخاب‌شده در Applied Steps نمایش داده می‌شود و می‌توان مستقیماً آن را ویرایش نمود.

Status Bar

در پایین صفحه، Status Bar قرار دارد که تعداد ستون‌ها و سطرهاى داده‌ی فعلی را نشان می‌دهد.

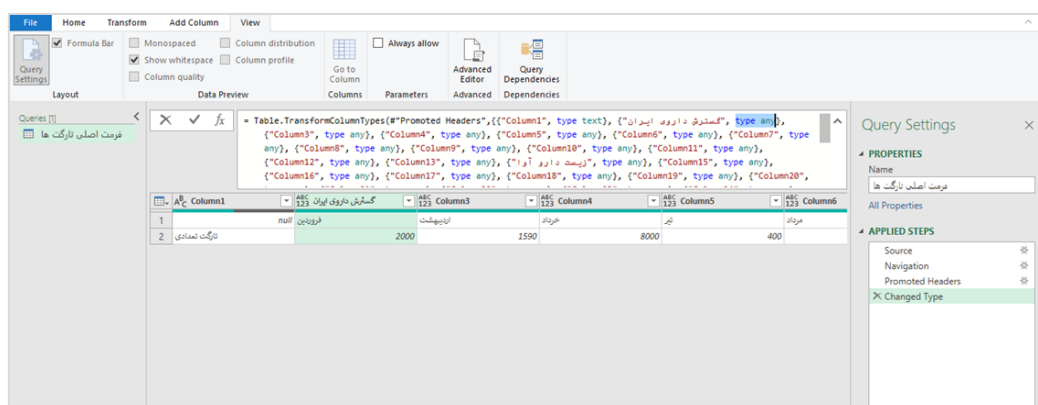
Data Type

در بالای هر ستون، آیکونی نمایش داده می‌شود که نوع داده را مشخص می‌کند:

- برای داده‌های متنی Whole Number 123
- برای داده‌های عددی Text A^B_C
- برای داده‌های ترکیبی $A^B_C 123$

Power Query به‌صورت خودکار نوع داده‌ها را تشخیص می‌دهد و در صورت لزوم آن‌ها را (Changed Type) می‌کند.

در Formula Bar نیز می‌توان دید که چه نوع داده‌ای برای هر ستون تعیین شده است.



مثلاً اگر داده‌ای Date/Time/Zone باشد، نمی‌توان آن را مستقیماً به Integer تبدیل کرد. برخی از تبدیل‌ها (Conversion) ممکن است باعث از دست رفتن بخشی از داده‌ها شوند، بنابراین باید با دقت انجام شوند. (تصویر از سایت powerquery.how)

Data Types	1.2	\$	1 ² 3	%	📅	📅	🕒	🌐	🕒	ABC	✖✓
1.2 Decimal number		⊖	⊖	✓	✓	⊖	⚠	+	✓	✓	✓
\$ Currency	✓		⊖	✓	✓	⊖	⚠	+	✓	✓	✓
1 ² 3 Whole number	✓	✓		✓	✓	✓	⚠	+	✓	✓	✓
% Percentage	✓	⊖	⊖		✓	✓	✓	+	✓	✓	✓
📅 Date/Time	✓	⊖	⊖	✓		⊖	⊖	+	⚠	✓	⚠
📅 Date	✓	✓	✓	✓	✓		⚠	+	⚠	✓	⚠
🕒 Time	✓	✓	✓	✓	+	⚠		+	⚠	✓	⚠
🌐 Date/Time/Zone	✓	⊖	⊖	✓	⊖	⊖	⊖		⚠	✓	⚠
🕒 Duration	✓	⊖	⊖	✓	⚠	⚠	⚠	⚠		✓	⚠
ABC Text	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
✖✓ True/False	✓	✓	✓	✓	⚠	⚠	⚠	⚠	⚠	✓	
✓ Conversion possible ⚠ Conversion fails with error + Conversion possible, adds detail to the original value ⊖ Conversion possible, loses detail of the original value											



- ✓ Conversion possible تبدیل امکان پذیر است.
- ⚠ Conversion fails with error تبدیل با Error از بین می رود.
- + Conversion possible, adds detail to the original value تبدیل امکان پذیر است، و به مقدار اصلی جزئیات اضافه می شود.
- ⊖ Conversion possible, loses detail of the original value تبدیل امکان پذیر است، جزئیات مقدار اصلی از بین می رود.

منابع آموزشی Power Query

چند منبع مفید برای یادگیری عمیق تر:

- سایت رسمی powerquery.how آموزش های انگلیسی از نویسنده هلندی الاصل
 - کتاب The Definitive Guide to Power Query
 - وبسایت Gorilla.bi شامل مقالات و آموزش های حرفه ای در زمینه Power Query
- در این منابع، نمودارهایی وجود دارد که نشان می دهد انواع داده (Data Types) چگونه به هم تبدیل می شوند و در چه مواردی باید مراقب از دست رفتن اطلاعات بود.
- بخش Data Preview یکی از مهم ترین قسمت های محیط Power Query Editor است. این بخش به ما کمک می کند داده ها را ببینیم، بررسی کنیم و قواعد پاک سازی را روی آن ها اعمال کنیم.

ویژگی های اصلی بخش Data Preview

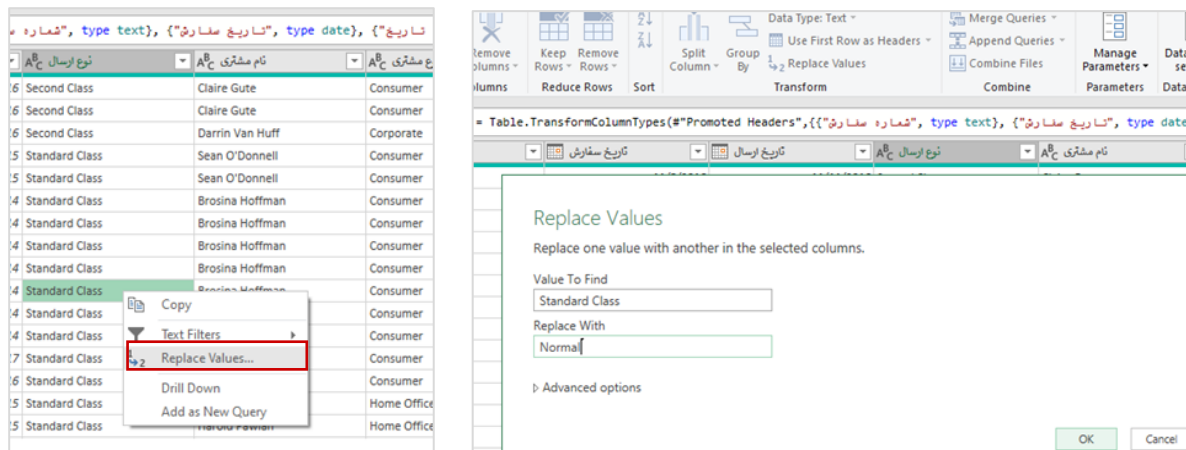
Lazy Load

در Power Query نیازی نیست کل داده را به صورت کامل بارگذاری کنیم. این ابزار فقط یک پیش نمایش از داده ها را نمایش می دهد تا بتوانیم قوانین پاک سازی و تغییر را تعریف کنیم. به همین دلیل، بارگذاری داده ها به صورت تدریجی و با تأخیر انجام می شود. وقتی اسکرول را به پایین می کشیم، ردیف های بیشتری از داده به صورت خودکار بارگذاری می شوند. مزیت این ویژگی این است که باعث افزایش سرعت کار و کاهش مصرف حافظه می شود، چون فقط بخش لازم از داده نمایش داده می شود.

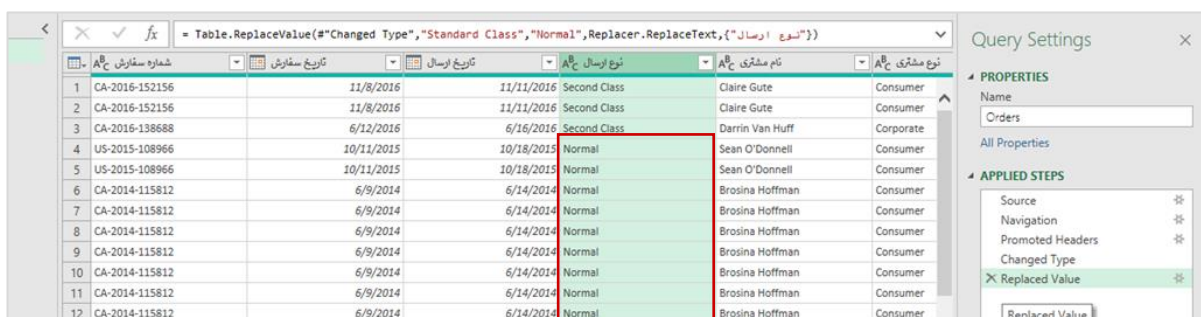
آنچه در Data Preview می بینیم، نسخه ی واقعی داده ها نیست؛ بلکه یک نمونه ی کوچک برای کار روی ساختار است. بنابراین هر تغییری که در Power Query انجام می دهیم، در واقع روی مراحل (Steps) اعمال می شود، نه روی داده ی واقعی. زمانی که داده ها Close & Load می شوند، تمام این مراحل روی کل داده ی اصلی اعمال می گردند.

Cells and errors

برخلاف اکسل، در محیط Power Query نمی توان سلول ها را به صورت دستی ویرایش کرد. در واقع ویرایش باید اصولی و قانونمند باشد، یعنی اگر داده ای اشتباه است و نیاز به تغییر دارد، باید از ابزارهای موجود در Power Query استفاده کنیم، مثل: Filter Rows/ Remove Columns/ Replace Values / Split Column.



با هر تغییر، یک Step جدید در بخش Applied Steps اضافه و ثبت می شود که می توان آن ها را حذف کرد،

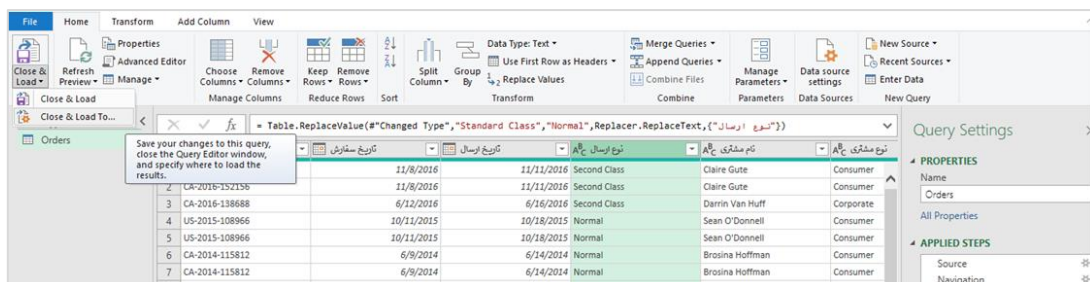


ترتیبشان را تغییر داد، یا مستقیماً فرمول M آن‌ها را در Formula Bar ویرایش نمود.



پس از انجام تغییرات در داده، باید آن را از Power Query به محیط مقصد برگردانیم. این کار از طریق دو گزینه انجام می‌شود:

- Close & Load، داده را به صورت پیش فرض به اکسل در قالب Table بارگذاری می‌کند.



- Close & Load To..., به ما اجازه می‌دهد انتخاب کنیم که داده در چه قالبی بارگذاری شود

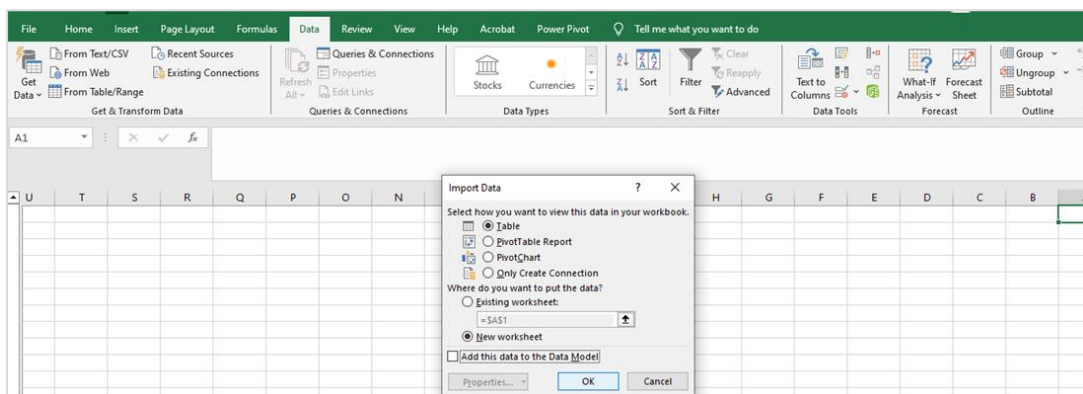


Table: در یک شیت اکسل جدول را ذخیره کند.

Pivot Table Report: فایل را آماده می کند برای اینکه گزارش Pivot بگیریم.

Pivot Chart: فایل را آماده می کند که گزارش نموداری بگیریم.

Only Create Connection: فقط یک ارتباط برقرار می کند که اگر جایی نیاز داشتیم از آن استفاده کنیم.

Existing Worksheet: می توان انتخاب کرد که چه شیت اکسلی داده را ببرد.

یا گزینه Add This Data To The Data Model که داده را به محیط Power Pivot می برد.

زمانی که داده های منبع تغییر کنند (مثلاً فایل اکسل به روزرسانی شود) با فشردن دکمه Refresh تمام مراحل

(Applied Steps) به صورت خودکار دوباره اجرا می شوند.

به این ترتیب، داده های جدید هم دقیقاً طبق همان قوانین تمیز و آماده سازی قبلی، پردازش خواهند شد. یعنی نیازی

نیست دوباره مراحل را از اول انجام دهیم و همانطور که گفته شد، Power Query یک محیط تکرارشونده و خودکار است.

با فشردن دکمه های Alt + F12 رو شیت اکسل دوباره به محیط Power Query میتوانیم ورود کنیم.

مدیریت خطاها و ابزارهای کاربردی در Power Query

هنگامی که داده ها از اکسل به Power Query منتقل می شوند، ممکن است برخی سلول ها دارای خطا (Error)

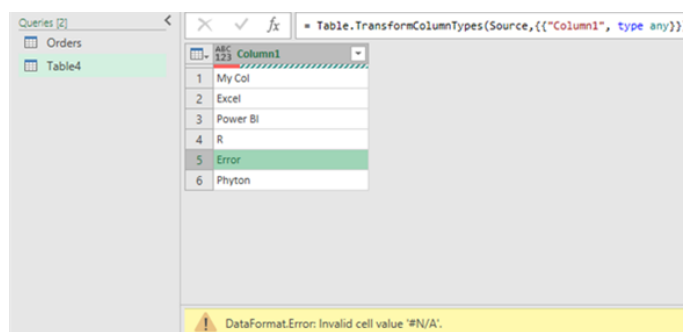
باشند. شناخت رفتار Power Query در این حالت و روش های رفع خطا از اصول پایه در پاک سازی داده است. در ادامه

مهم ترین نکات و ابزارهای مرتبط توضیح داده می شوند

اگر در فایل اکسل داده یا مقداری اشتباه در یک سلول تایپ شده باشد (مثلاً به جای عدد، عبارت نامعتبر نوشته شده)،

در Power Query آن سلول به صورت Error نمایش داده می شود. بالای ستون مربوطه، کمی قرمز رنگ ظاهر می شود. با

کلیک روی سلول، در پایین صفحه جزئیات خطا نمایش داده می شود و علت Error مشخص می شود.

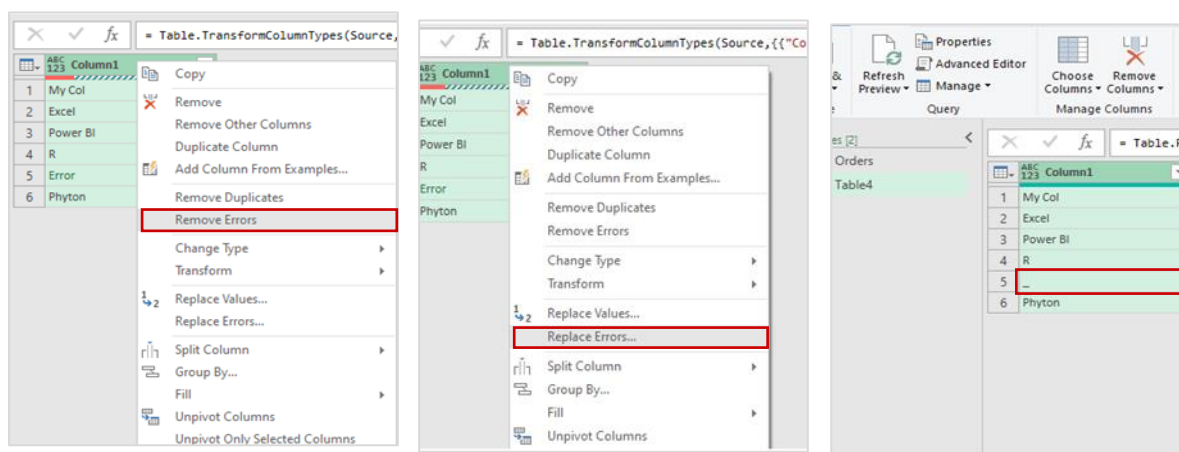


نکته: نباید روی کلمه Error کلیک کنیم، چون Power Query در این حالت، فقط همان ردیف را نگه می‌دارد و سایر داده‌ها را حذف می‌کند.

برای رفع خطا دو روش اصلی وجود دارد:

الف) حذف ردیف‌های دارای خطا: روی سلول یا ستون موردنظر راست کلیک کرده و گزینه‌ی Remove Errors را انتخاب می‌کنیم. با این کار، تمام ردیف‌هایی که دارای Error هستند حذف می‌شوند و البته دقت داشته باشیم که بقیه‌ی داده‌ی آن سطر نیز از بین می‌رود.

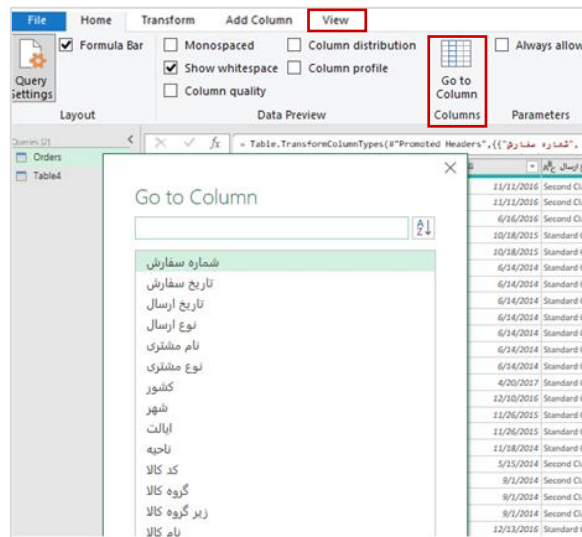
ب) جایگزین کردن خطا با مقدار دلخواه: روی ستون راست کلیک کرده و گزینه‌ی Replace Errors را انتخاب می‌کنیم. در این حالت می‌توانیم تعیین کنیم که به جای خطا چه چیزی نمایش داده شود؛ مثلاً یک خط تیره(-)، عدد صفر، یا عبارت "Unknown". انتخاب مقدار جایگزین مربوط به نوع کسب و کار شما و نیاز مخاطب است.



در Power Query، برخلاف اکسل، نمی‌توان ردیف‌ها را «به صورت دستی» حذف کرد. باید مشخص کنیم چه داده‌ای را نمی‌خواهیم، (مثلاً همه‌ی سلول‌هایی که مقدارشان «-» (خط تیره) است) که برای این کار از ابزار Filter استفاده می‌کنیم و انتخاب می‌کنیم که کدام ردیف‌ها را نیاز نداریم.

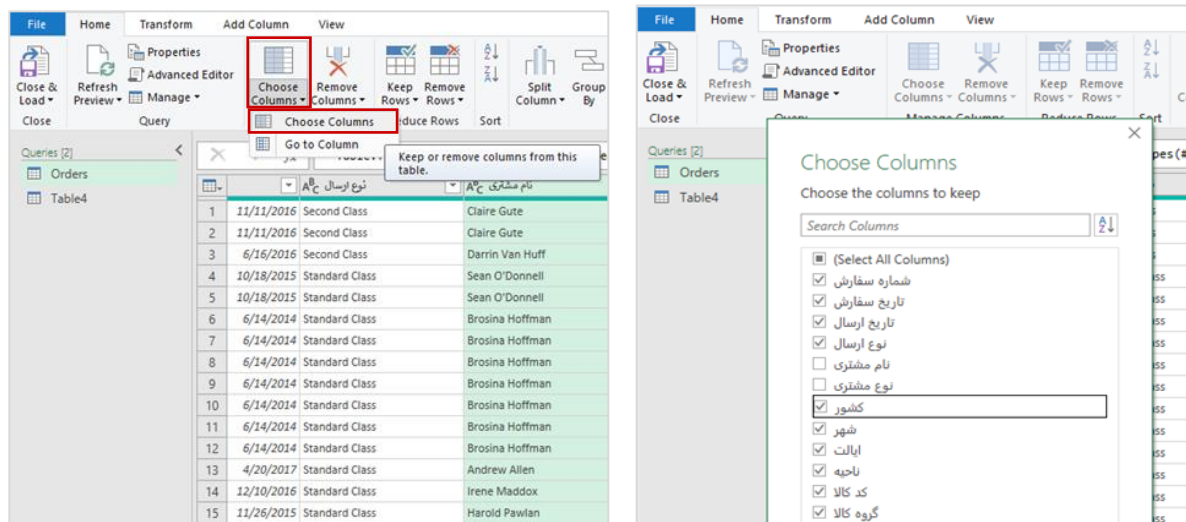
Go to Column

زمانی که تعداد ستون‌ها زیاد است، از گزینه‌ی Go to Column در تب View استفاده می‌کنیم. با انتخاب نام ستون، مستقیماً به همان ستون هدایت می‌شویم. این قابلیت برای مرور سریع در جداول بزرگ بسیار مفید است.



Choose Columns

با این گزینه می‌توانیم مشخص کنیم کدام ستون‌ها را نگه داریم و بقیه را حذف کنیم، این ابزار برای ساده‌سازی داده و کاهش حجم Preview بسیار کاربردی است.

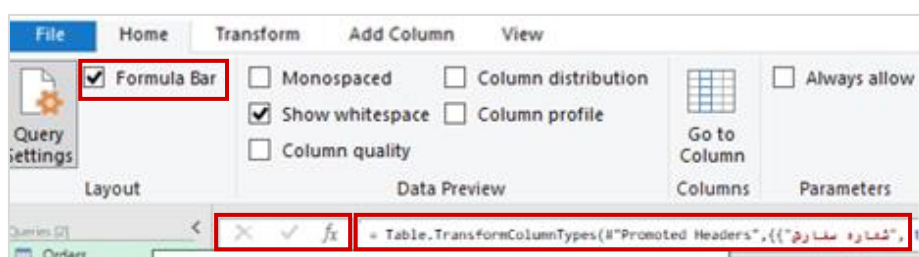


به جای اینکه با اکسل این ستون‌ها را مخفی یا حذف کنیم، در اینجا با گزینه Choose Columns انتخاب می‌کنیم که فعلاً چه ستون‌هایی را نیاز نداریم و تمام این تغییرات (حذف، جایگزینی، فیلتر و انتخاب ستون‌ها) در بخش Applied Steps ذخیره می‌شوند.

اگر فایلی را در اکسل ذخیره کنیم، این تغییرات به صورت خودکار در خروجی اعمال می شوند. اما اگر بخواهیم به حالت اولیه برگردیم، می توانیم Step های مربوطه را حذف کنیم تا داده به شکل اصلی بازگردد. در واقع Power Query یک واسطه (Interface) بین منبع داده و فایل خروجی است. داده ای اصلی دست نخورده باقی می ماند و فقط مراحل پردازش در Power Query ثبت می شوند.

Formula Bar

در تب View می توان Formula Bar را فعال کرد. در این بخش، فرمول مربوط به هر Step به زبان M نمایش داده می شود. می توان این فرمول را ویرایش کرد یا با کلیک روی دکمه fx یک Step جدید ایجاد نمود و فرمول موردنظر را نوشت. برای مثال، می توان با اضافه کردن Step جدید، محاسبه ای خاص یا جایگزینی پیشرفته را به داده افزود.



این بخش شامل مجموعه ای از ابزارها و قابلیت هایی است که برای بررسی، ویرایش و مدیریت ستون های داده در محیط Power Query Editor استفاده می شود و شامل ۱۰ قابلیت اصلی می باشد.

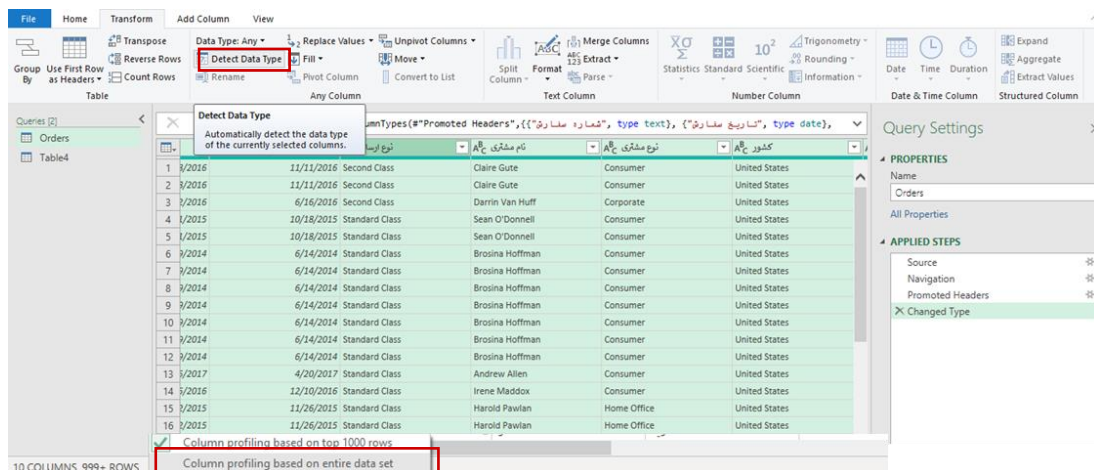
Basic Column Actions

Detect Data Type

زمانی که تعداد ستون ها زیاد است و می خواهیم نوع داده ای هر ستون را بررسی یا تصحیح کنیم، از ابزار Detect Data Type استفاده می کنیم. با کلیدهای Ctrl + A تمام ستون ها را انتخاب می کنیم و از مسیر:

Transform → Any Column → Detect Data Type، به صورت خودکار نوع داده ای هر ستون تشخیص داده می شود.

به صورت پیش فرض، این تشخیص بر اساس ۱۰۰۰ سطر اول داده ها انجام می شود. اما می توان این تنظیم را از پایین صفحه (Status Bar) تغییر داد و گزینه ای Entire Dataset را انتخاب کرد تا تشخیص نوع داده بر اساس تمام ردیف ها انجام شود.

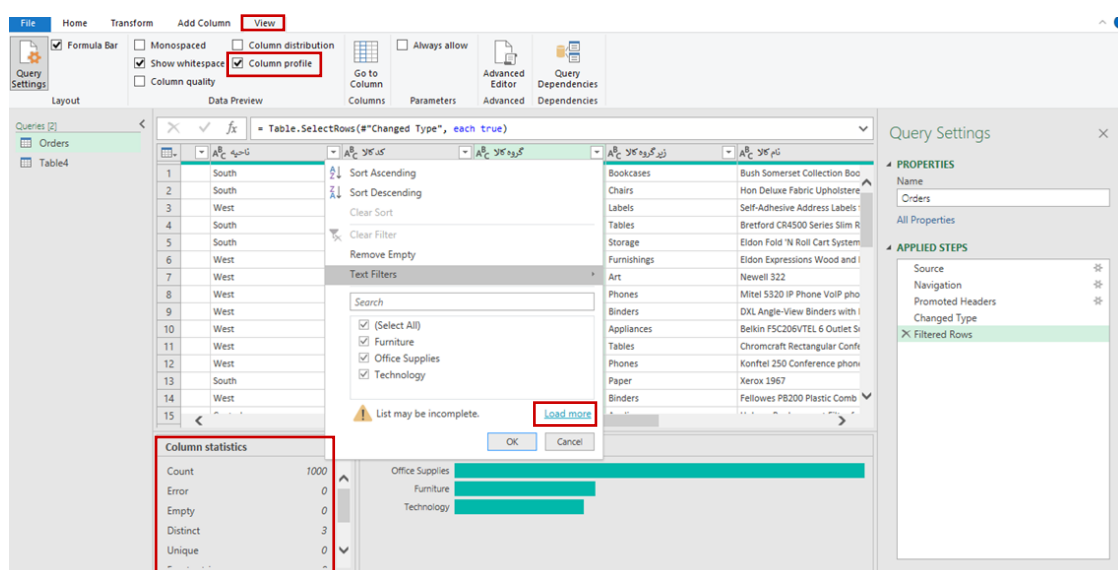


Column Profile

در تب View، با فعال کردن گزینه‌ی Column Profile، در پایین صفحه خلاصه‌ای از اطلاعات ستون انتخاب‌شده نمایش داده می‌شود. اطلاعاتی که در این بخش نمایش داده می‌شوند عبارت‌اند از:

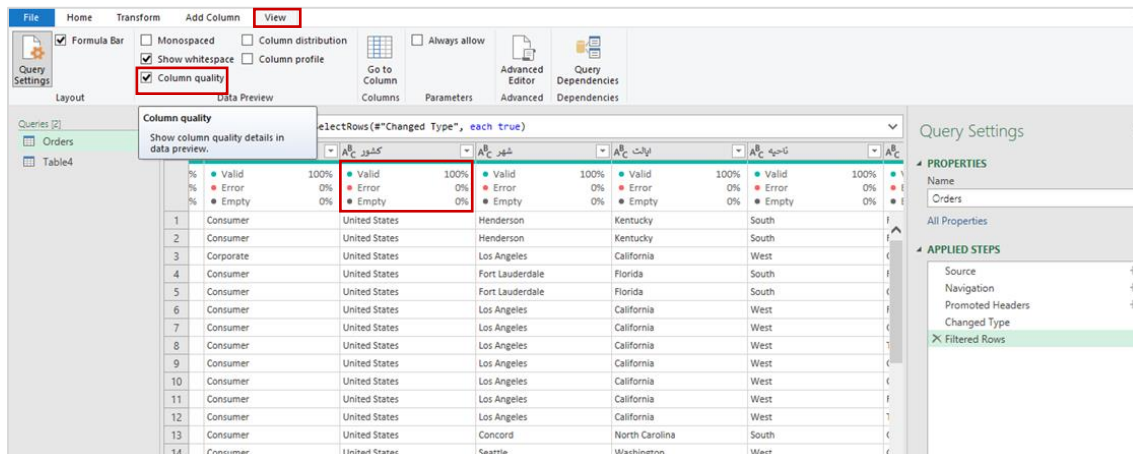
- تعداد کل ردیف‌ها (Count)
- تعداد مقادیر Error
- Max و Min
- تعداد مقادیر متمایز (Distinct) و تک‌نمونه‌ای (Unique)
- نمودار میله‌ای (Bar Chart) برای نمایش توزیع داده‌ها

اگر داده‌ها زیاد باشند، می‌توان روی گزینه‌ی Load More کلیک کرد تا تمام داده‌ها بررسی و خلاصه‌سازی شوند



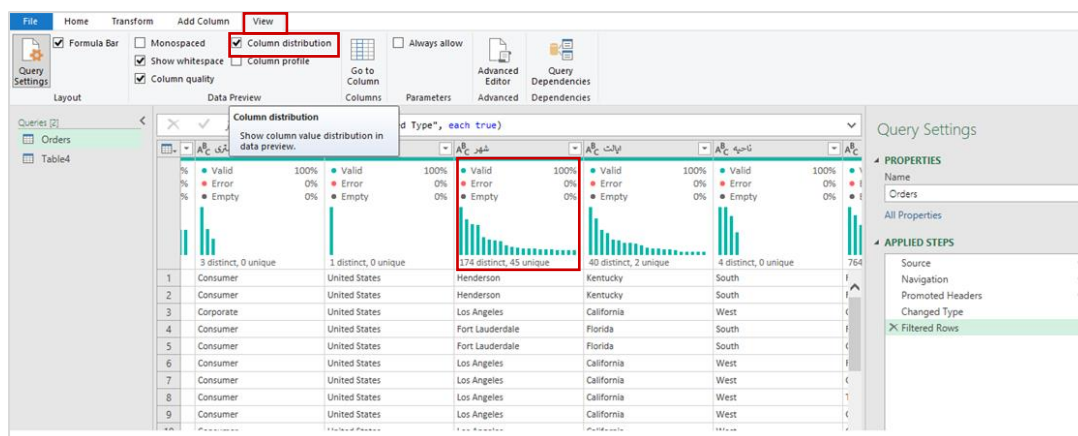
Column Quality

در تب View گزینه‌ی Column Quality را فعال کنید. این ابزار درصد داده‌های Valid، Error و Empty را برای هر ستون نمایش می‌دهد. به کمک آن می‌توان ستون‌هایی را که دارای داده‌های ناقص یا خطا دارند سریع شناسایی کرد.



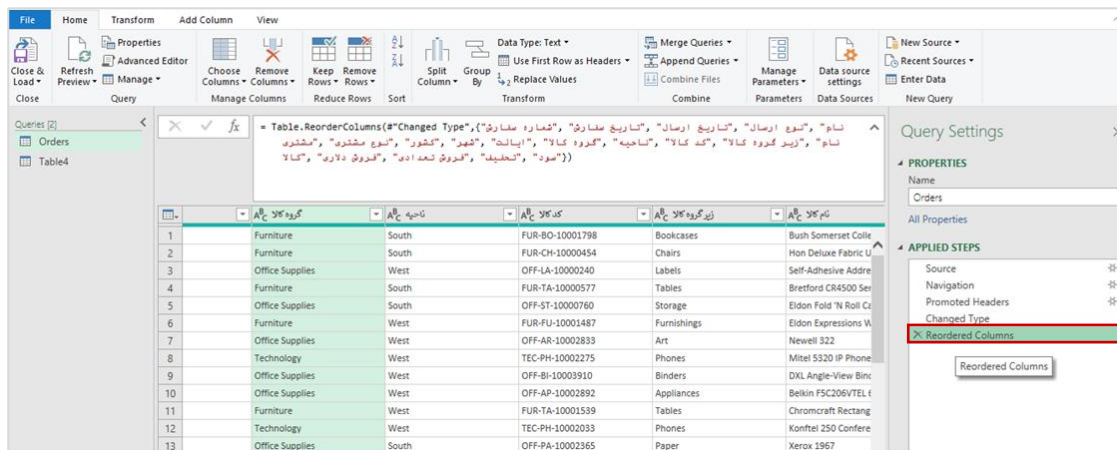
Column Distribution

در تب View می‌توان گزینه‌ی Column Distribution را فعال کرد. این بخش، توزیع مقادیر هر ستون را به شکل نمودار میله‌ای نمایش می‌دهد و به ما کمک می‌کند متوجه شویم داده‌ها چگونه پخش شده‌اند (تکراری، یکنواخت، یا دارای مقادیر پرت).



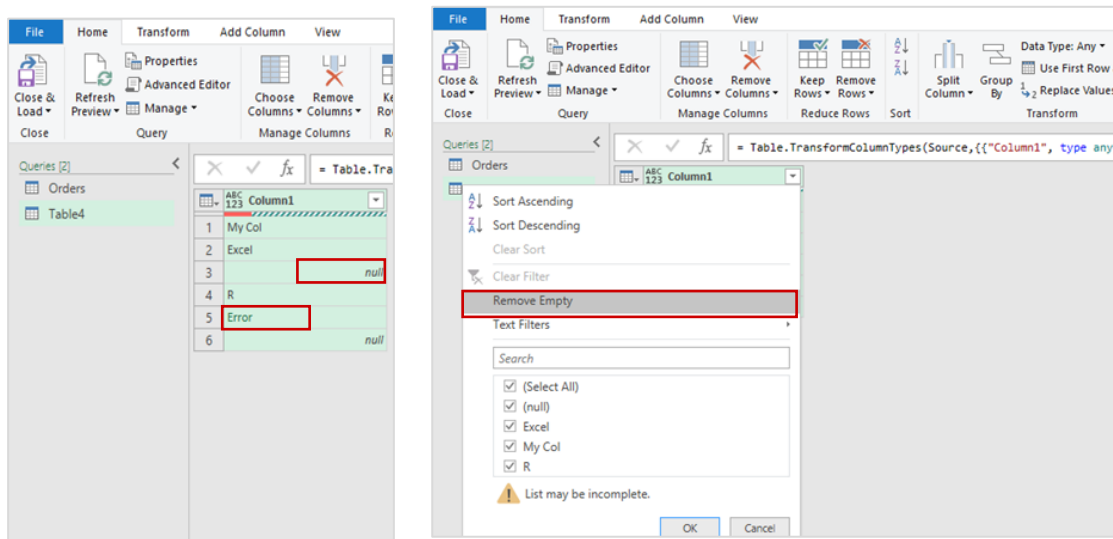
Reorder Columns

برای جابه‌جایی ستون‌ها کافی است ستون را با Drag & Drop به محل دلخواه بکشیم. با انجام این عمل، یک Step جدید با نام Reorder Columns در بخش Applied Steps ایجاد می‌شود. Power Query برای این عمل به صورت خودکار یک فرمول M تولید می‌کند که ترتیب جدید ستون‌ها را مشخص می‌کند.



Sort & Filter

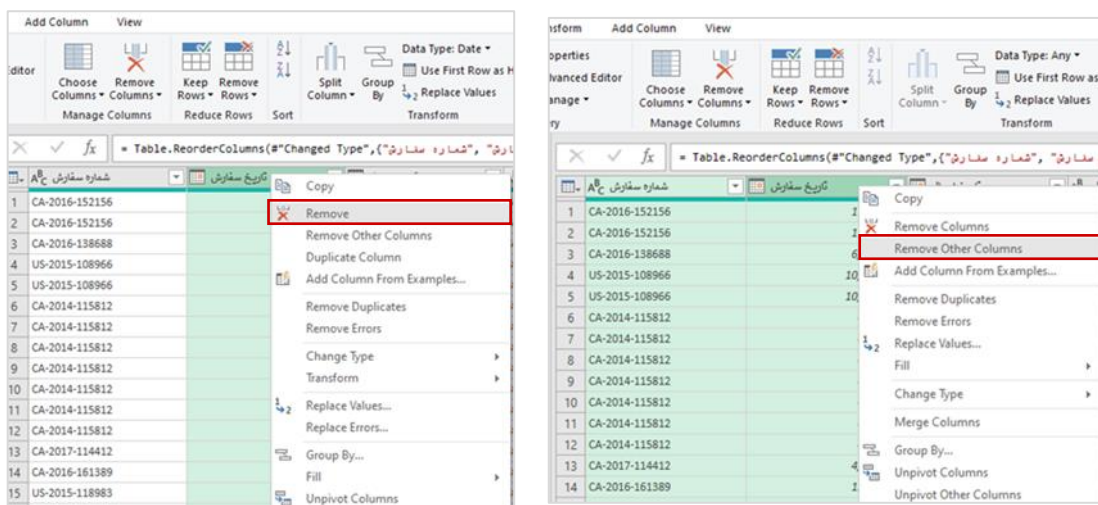
عملکرد این بخش مشابه اکسل است. می‌توان داده‌ها را به صورت Ascending (صعودی) یا Descending (نزولی) مرتب کرد یا فیلترهای مختلف اعمال نمود. البته باید به این نکته توجه کرد که هر عمل Sort یا Filter یک Step جدید در Applied Steps اضافه می‌کند. اگر ضرورتی ندارد، از انجام مرتب‌سازی یا فیلترهای غیرضروری خودداری کنید. در فیلترها گزینه‌ای به نام Remove Empty وجود دارد که باعث حذف مقادیر هم Null و هم Blank از داده‌ها می‌شود، که انجام آن بهتر از برداشتن تیک Null در فیلتر است.



Remove / Remove Other Columns

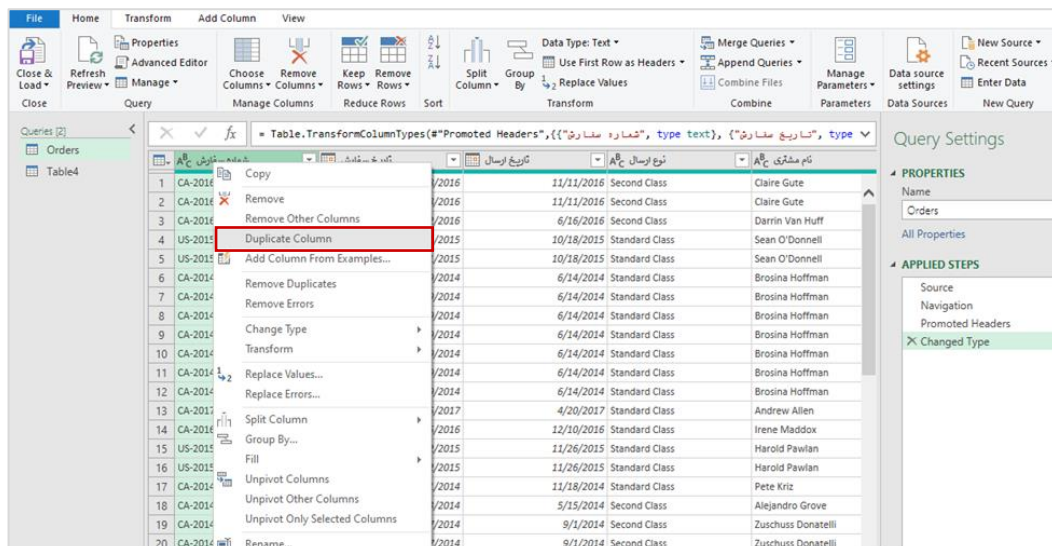
اگر بخواهیم برخی ستون‌ها را حذف کنیم روی ستون مورد نظر راست کلیک و گزینه Remove Columns را انتخاب می‌کنیم.

اگر بخواهیم فقط چند ستون خاص را نگه داریم آن ستون‌ها را با Ctrl یا Shift انتخاب کرده، سپس راست کلیک و گزینه Remove Other Columns را انتخاب می‌کنیم. این قابلیت مخصوص زمانی است که تعداد ستون‌ها زیاد است و می‌خواهیم فقط چند ستون کلیدی باقی بمانند.



Duplicate Column

برای ساخت یک نسخه‌ی تکراری از ستون، از گزینه Duplicate Column استفاده می‌کنیم. Power Query ستون جدید را در انتهای جدول ایجاد می‌کند. می‌توان از این ستون کپی برای انجام تغییرات خاص استفاده کرد، مثلاً اگر ستون اصلی شامل تاریخ است، در ستون تکراری فقط سال یا ماه را نمایش دهیم.

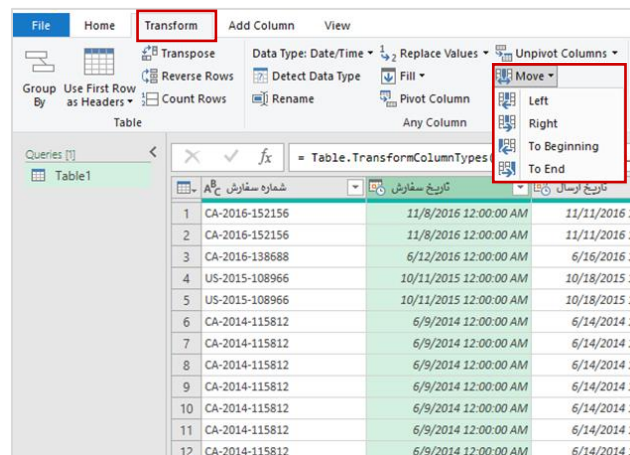


Rename Columns

برای تغییر نام ستون کافی است روی Header آن دوبار کلیک کنیم یا از گزینه‌ی Rename استفاده نماییم. هر بار که نام ستونی تغییر کند، یک Step جدید به نام Renamed Columns در Applied Steps ایجاد می‌شود.

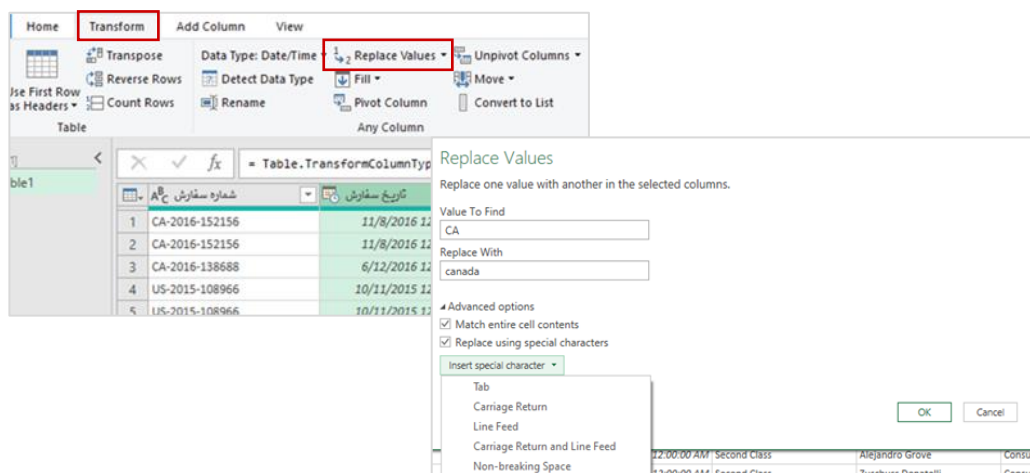
Move Columns

در تب Transform گزینه‌هایی برای Move Left / Right / To Beginning / To End وجود دارد که برای جابجایی سریع ستون‌ها بدون Drag & Drop استفاده می‌شوند. (این امکان در گزینه‌های کلیک راست نیز وجود دارد)



Replace Value

با این گزینه می‌توان هر داده‌ای را با هر عبارت دیگری جایگزین کرد. یک بخش Advanced Option هم دارد که امکانات دیگری هم دارد. مثل Match Entire Cell Content که اگر این تیک این گزینه را فعال کنیم باید کل سلول حاوی آن کلمه یا Value to find باشد. گزینه بعدی هم Replace Using Special ، که می‌خواهیم کاراکترهای خاصی جایگزین بشوند که از کادر پایین‌تر، کاراکتر مورد نظر را انتخاب می‌کنیم.





Copy

اگر روی سرستون کلیک راست کنیم و Copy را بزنیم، اگر سطرها زیاد باشند فقط هزار سطر اول کپی می‌شوند، نه همه‌ی سطرها و اگر در یک برنامه دیگر مثلاً Notepad یا Word اضافه کنیم، فقط هزار سطر اول کپی می‌شود و نمایش داده می‌شود.

نکات تکمیلی مربوط به Applied Steps :

- Step ها قابلیت جابه‌جایی دارند. در برخی کارها این جابه‌جایی‌ها خطرناک هستند و اگر وابستگی به هم داشته داشته باشند خطا می‌دهند. پس نیازمند دقت است که آیا این جابه‌جایی‌ها درست است یا نه، یعنی آیا تقدم و تأخر باهم دارند یا خیر.
- Step هایی که در کنارشان علامت چرخ‌دنده دارند یعنی ویرایش بصری یا UI Editing دارند، و آنهایی که ندارند فقط Formula Bar این قابلیت را به ما می‌دهد.
- اگر روی Step آخر باشیم و هر اقدام جدیدی انجام بدهیم، یک Step جدید اضافه می‌شود. اما اگر مثلاً روی Step سوم باشیم و بخواهیم یک اقدام جدید انجام بدهیم — مثلاً یک ستون را Remove کنیم — یک پیغام می‌دهد که آیا می‌خواهید Step جدید را در این محل انجام بدهیم، که در این صورت ممکن است Step های بعدی دچار مشکل شوند.

جلسه دوم

دستورات پیشرفته ستون‌ها

Fill Up, Fill Down

فرض کنید ما جدولی داریم که سطرهایش Merge شده باشند، که باید این مورد را اصلاح کنیم. در اکسل این مسئله با دو روش فرمول نویسی و هم بدون فرمول نویسی حل می‌شود.

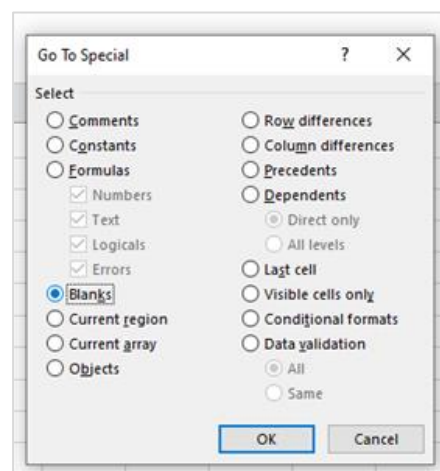
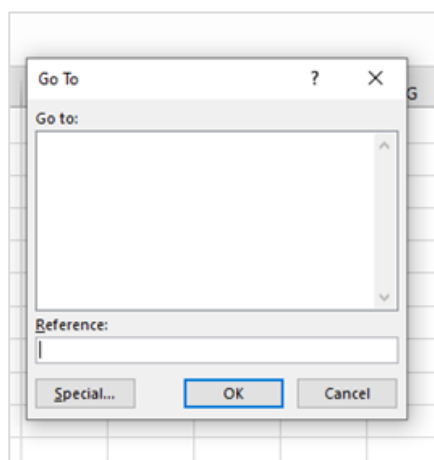
اگر سطرها را Unmerge کنیم، داده فقط در سلول اول می‌ماند و بقیه سلول‌ها خالی می‌شوند و اگر بخواهیم از این جدول یک Pivot Table بگیریم، سلول‌های Blank خواهیم داشت، در حالی که سلول‌های بعدی داده دارند، اما خود سلول در ستون Merged خالی است. بنابراین اگر گزارشی از داده‌ها بگیریم، نتیجه اشتباه خواهد بود.

D	C	B	A
مبلغ	تاریخ	نام شرکت	وضعیت
۱۲۰,۰۰۰	۱۳۹۸/۰۲/۰۱	الف	تهیهات پرداختی
۴۲۰,۰۰۰	۱۳۹۸/۰۲/۰۲	ب	
۱۲۰,۰۰۰	۱۳۹۸/۰۲/۱۲	ب	
۳۰۰,۰۰۰	۱۳۹۸/۰۲/۱۸	ت	
۱۳۰,۹۰۰	۱۳۹۸/۰۲/۱۰	ث	صدور لایحه
۱۲۰,۰۰۰	۱۳۹۸/۰۲/۰۱	الف	
۲۳۰,۰۰۰	۱۳۹۸/۰۲/۱۱	ج	مطرح در ارگان اعتباری
۳۱۰,۰۰۰	۱۳۹۸/۰۲/۲۰	الف	
۳۴۰,۰۰۰	۱۳۹۸/۰۲/۱۰	ت	
۱۲۰,۰۰۰	۱۳۹۸/۰۲/۰۸	ج	



• راه‌حل‌ها در اکسل

یکی از روش‌ها این است که ستون را انتخاب کنیم، Unmerge کنیم، سپس Ctrl + G بزنیم تا به کادر Go To برویم. در آنجا گزینه Special را انتخاب می‌کنیم، تیک Blank را می‌زنیم و OK می‌کنیم.



سپس، بدون اینکه جایی کلیک کنیم، شروع به تایپ می‌کنیم: در سلول خالی علامت مساوی (=) می‌گذاریم و سلول بالایی را انتخاب می‌کنیم مثلاً (A2) و سپس **Ctrl + Enter** می‌زنیم تا آنچه نوشتیم در همه سلول‌های دیگر هم به صورت نسبی اعمال شود. بعد از ستون **Copy** می‌گیریم و روی خودش **Paste Value** می‌زنیم تا فرمول‌ها به مقادیر تبدیل شوند. این کار باعث می‌شود اگر جدول **Sort** شد یا جابه‌جا شد، فرمول‌ها به هم نریزند.

وضعیت	نام شرکت	تاریخ	مبلغ
تسهیلات پرداختی	الف	۱۳۹۸/۰۳/۰۱	۱۲۰,۰۰۰
تسهیلات پرداختی	ب	۱۳۹۸/۰۳/۰۲	۲۲۰,۰۰۰
تسهیلات پرداختی	ب	۱۳۹۸/۰۳/۱۲	۱۲۰,۰۰۰
تسهیلات پرداختی	ت	۱۳۹۸/۰۳/۱۸	۳۰۰,۰۰۰
تسهیلات پرداختی	ث	۱۳۹۸/۰۳/۱۰	۱۳۰,۹۰۰
صدور ابلایه	الف	۱۳۹۸/۰۳/۰۱	۱۲۰,۰۰۰
صدور ابلایه	ج	۱۳۹۸/۰۳/۱۱	۲۲۰,۰۰۰
مطرح در ارگان اعتباری	الف	۱۳۹۸/۰۳/۲۰	۳۱۰,۰۰۰
مطرح در ارگان اعتباری	ت	۱۳۹۸/۰۳/۱۰	۲۴۰,۰۰۰
مطرح در ارگان اعتباری	ج	۱۳۹۸/۰۳/۰۸	۱۲۰,۰۰۰

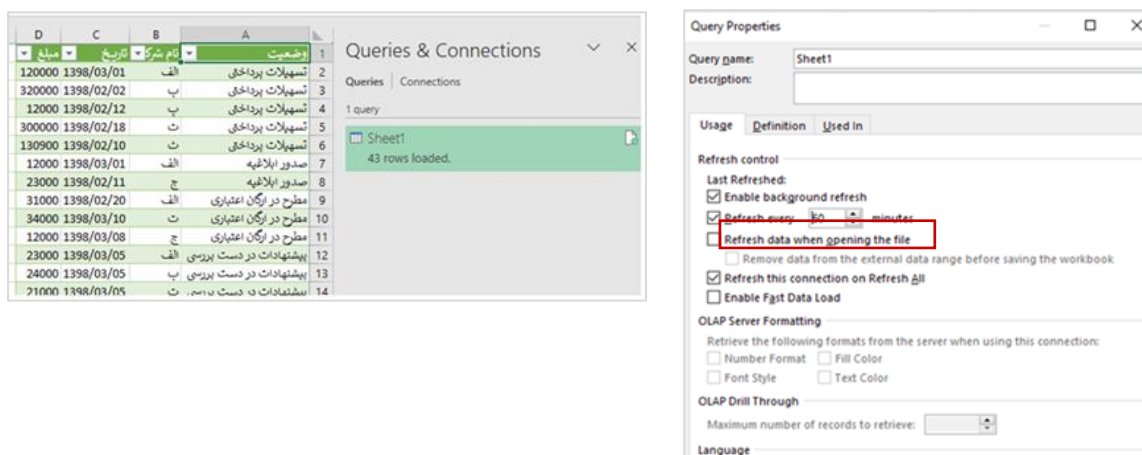
اما این روش **Dynamic** نیست و اگر داده‌ها دوباره اضافه شوند، باید این مراحل را تکرار کنیم. تکرار کارها یکی از مواردی است که **Power Query** آن را برای ما حذف می‌کند و مثل ماشین زمان عمل می‌کند و کار را جلو می‌اندازد.

• راه‌حل در Power Query:

روی ستون کلیک راست می‌کنیم و گزینه **Fill Down** را انتخاب می‌کنیم. این کار مقادیر **Null** را با مقدار بالاسری‌شان پر می‌کند، یا **Fill Up** که با مقادیر پایین‌تر پر می‌کند. انجام این کار آسان‌تر است و برای همیشه در فایل ذخیره می‌شود.

The first screenshot shows a Power Query table with columns 'وضعیت' (Status) and 'تاریخ' (Date). The 'وضعیت' column contains various status names, and the 'تاریخ' column contains dates. Some cells in the 'وضعیت' column are null. The second screenshot shows the context menu for the 'وضعیت' column, with the 'Fill' option highlighted. The third screenshot shows the 'Fill Down' option selected in the submenu.

در اکسل هم می‌توانیم آن را Refresh کنیم. در بخش Queries & Connections گزینه Refresh وجود دارد، و حتی اگر روی Query در اکسل کلیک راست کنیم و Properties را بزنیم، می‌توانیم زمان Refresh را تنظیم کنیم. در قسمت Refresh Every ... مشخص می‌کنیم که مثلاً هر چند دقیقه یکبار داده‌ها به‌روز شوند.



نکته قابل تامل این است که فقط Fill Down و Fill Up داریم، چون این سیستم‌ها ساختار جدولی و عمودی دارند. اما Fill Right و Fill Left نداریم. پس اگر سلولی با سلول کناری‌اش Merge شده بود، راه‌حل آن است که جدول را ابتدا بچرخانیم، Fill Down انجام دهیم و دوباره آن را برگردانیم تا به شکل افقی نمایش داده شود. (این مورد در درس‌های بعدی توضیح داده خواهد شد.)

Remove Duplicate

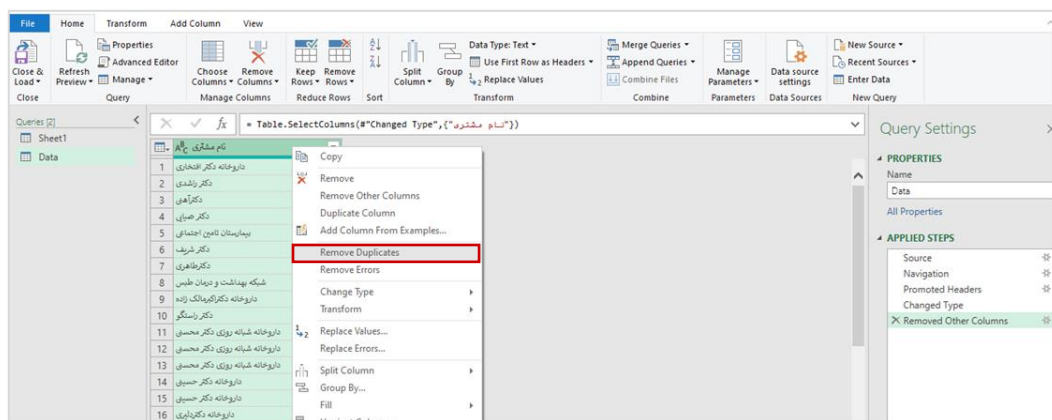
گاهی در کار پاکسازی نیاز به یک فهرست خالص (Unique List) از داده‌ها داریم. در نسخه‌های قبل از اکسل 2021، به‌دست آوردن یک فهرست خالص از مقادیر یک ستون به‌صورت داینامیک بسیار چالش‌برانگیز بود و با فرمول‌های آرایه‌ای سخت و پیچیده انجام می‌شد. اما در اکسل 2021 به‌بعد، با تابع UNIQUE این مشکل برطرف شده است.

در Power BI نیز خیلی اوقات ما Dimension‌هایی داریم که ممکن است بدون اطلاع ما داده‌های تکراری در آن‌ها وجود داشته باشد. این داده‌های تکراری ممکن است هنگام ورود داده ایجاد شوند و Data Model را دچار مشکل کرده و باعث خطا یا مشکلات تحلیلی برای کاربر شوند. در واقع، گاهی مشکل از داده‌ها هم می‌باشد و همیشه فقط بحث طراحی نیست. یکی از بهترین ابزارها برای حل این مسئله Power Query است. برای به‌دست آوردن فهرستی خالص و بدون تکرار از یک ستون، مراحل زیر را انجام می‌دهیم:

- روی ستون موردنظر کلیک راست می‌کنیم.
- ابتدا Remove Other Columns را می‌زنیم تا فقط همان ستون باقی بماند.

- سپس **Remove Duplicates** را انتخاب می‌کنیم تا مقادیر تکراری حذف شوند، حتی اگر در حال حاضر

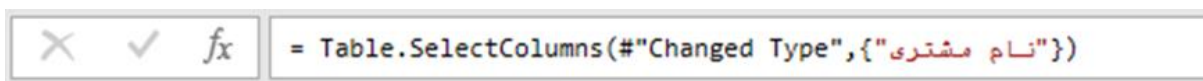
داده تکراری نداشته باشیم.



اگر در Power BI می‌خواهیم جدولی داشته باشیم که در یک رابطه‌ی One-to-Many (یک به چند) در سمت «One» قرار بگیرد، باید مطمئن شویم که هیچ‌گاه داده‌ی تکراری در آن وجود نداشته باشد. برای همین باید در Power Query قبل از بارگذاری داده، آن را تمیز کنیم.

گاهی ما فقط می‌دانیم گزینه‌ی Remove Duplicates وجود دارد، اما نمی‌دانیم چه زمانی و چرا باید از آن استفاده کنیم. ممکن است در ابتدا داده‌ی تکراری نداشته باشیم، اما وقتی داده‌های جدید اضافه می‌شوند، موارد تکراری به‌وجود بیایند. اگر این مرحله را از ابتدا انجام داده باشیم، از این مشکل جلوگیری می‌شود. حتی اگر ستونی در حال حاضر داده‌ی تکراری نداشته باشد، باز هم بهتر است روی آن Remove Duplicates اعمال کنیم، چون این مرحله در آینده از بروز خطا جلوگیری می‌کند. به این ترتیب، با این اقدام همیشه یک ستون یا فهرست غیرتکراری و تمیز خواهیم داشت و اگر در آینده داده‌ی تکراری هم اضافه شود، سیستم آن را تشخیص داده و حذف می‌کند.

(هر مرحله در Power Query در واقع یک فرمول دارد که با زبان M نوشته می‌شود. به توابع دقت کنید، برای جلسات بعدی)

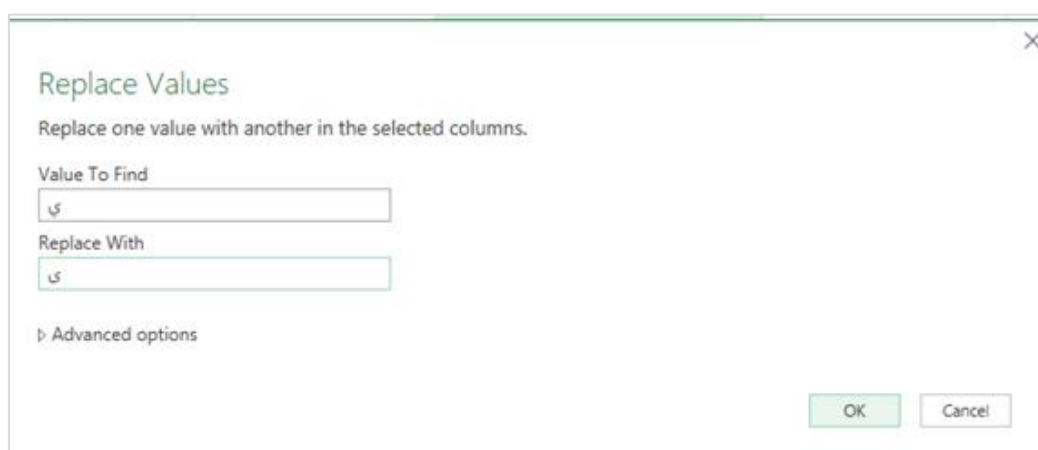


قبل از اجرای Remove Duplicates بهتر است چند مرحله پاکسازی دیگر هم مانند Replace Values و Transform انجام دهیم.

Replace Value: جایگزینی حروف یا نشانه‌هایی که ظاهرشان یکی است، ولی در سیستم متفاوت‌اند. مثلاً حرف

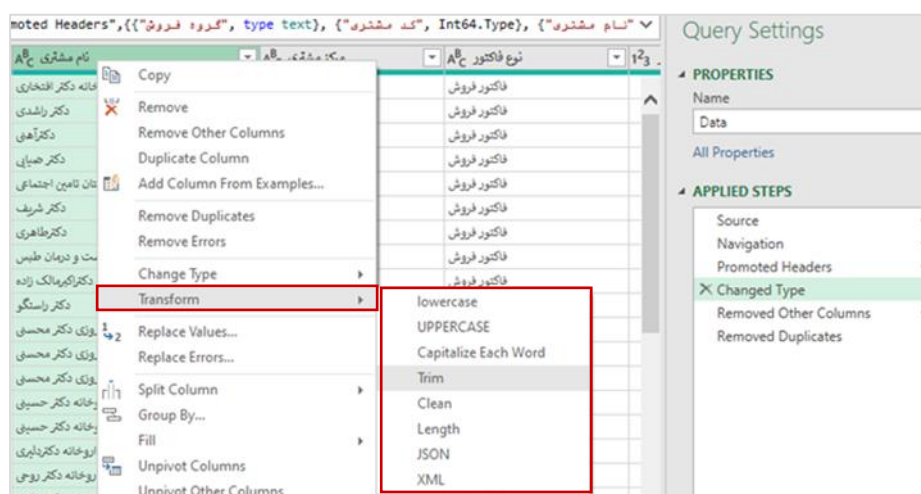
"ی" و "ك" عربی با "ی" و "ك" فارسی از نظر کامپیوتر دو کاراکتر متفاوت محسوب می‌شوند، در نتیجه Remove Duplicates آن‌ها را تکراری تشخیص نمی‌دهد. برای رفع این موضوع باید از Replace Value استفاده کنیم.

برای تایپ "ك" عربی در حالت کیبورد فارسی، کلیدهای Alt + 0223 را نگه دارید و سپس رها کنید، سپس در بخش جایگزینی، آن را با "ك" فارسی جایگزین کنید. این مرحله باعث می‌شود داده‌ها از نظر ظاهری و ساختاری یکسان شوند، و در نتیجه، عملیات Remove Duplicates به‌درستی کار کند. هر Replace جدیدی که انجام می‌دهیم، یک Step جدید به مراحل Power Query اضافه می‌کند و داده‌ها را از نظر ساختاری یکسان و استاندارد نگه می‌دارد.



برای دستور Transform نیز گزینه‌های مختلف وجود دارد مانند: Trim برای حذف فاصله‌های ابتدایی و انتهایی

داده‌ها، Clean، برای حذف کاراکترهای غیرقابل چاپ و ...

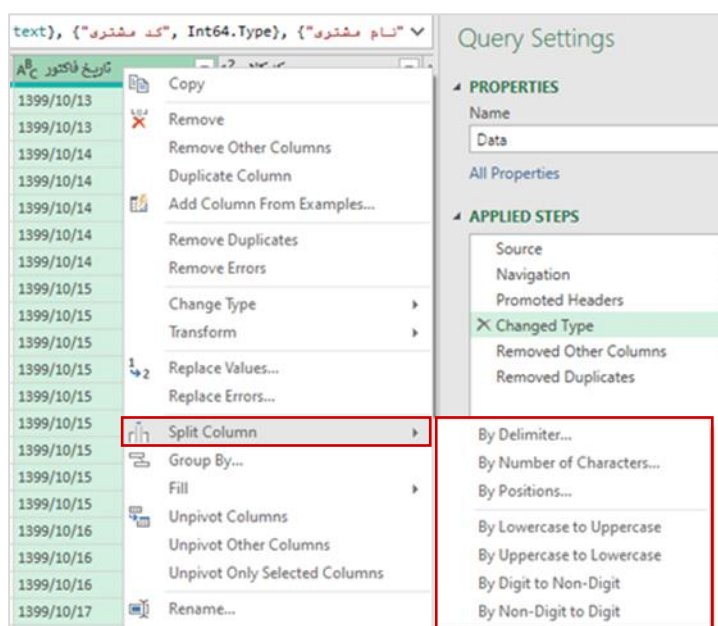


Split to Column

گاهی ستونی وجود دارد که داده‌های درون آن را به دلایل مختلف باید تفکیک (Split) شود. مثلاً ستونی از تاریخ‌ها شامل روز، ماه و سال که باید جدا شوند. گزینه‌ی Split to Column در Power Query ستون را می‌شکند و آن را به چند ستون تفکیک می‌کند. این دستور معادل تابع TEXTSPLIT در اکسل یا دستور Text to Columns است.

وقتی روی ستون کلیک راست می‌کنیم و گزینه Split Column را می‌زنیم، هفت روش مختلف برای جداسازی

(Split) پیشنهاد می‌دهد:



۱. By Delimiter : براساس جداکننده (Delimiter) مثل , , - , / یا Space

۲. By Number of Characters : براساس تعداد کاراکترها (مثلاً هر ۴ کاراکتر یک‌بار)

۳. By Position : براساس موقعیت یا جایگاه مشخص در متن

۴. By Lowercase to Uppercase : جداسازی بین حروف کوچک و بزرگ

۵. By Uppercase to Lowercase : برعکس مورد بالا

۶. By Digit to Non-Digit : جداسازی بین عدد و غیرعدد

۷. By Non-Digit to Digit : جداسازی بین غیرعدد و عدد

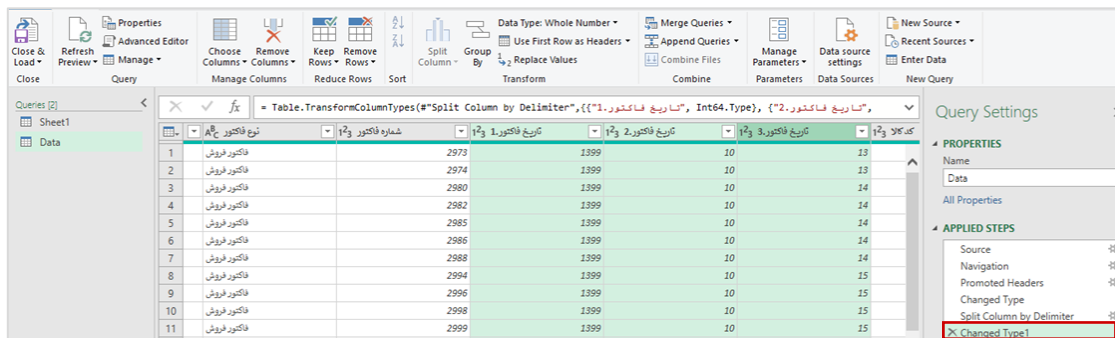
در روش By Delimiter، Power Query معمولاً خودش Delimiter را تشخیص می‌دهد مثلاً در تاریخ (/)

وقتی گزینه‌ی By Delimiter را انتخاب می‌کنیم، بخشی با عنوان Split At نمایش داده می‌شود که سه انتخاب دارد:

- Left-most Delimiter: جداکردن بر اساس اولین جداکننده
 - Right-most Delimiter: جداکردن بر اساس آخرین جداکننده
 - Each Occurrence of the Delimiter: جداکردن بر اساس تمام جداکننده‌ها
- نکته: نباید بگوییم «چپ‌ترین / راست‌ترین»، بلکه درست‌تر است بگوییم «اولین» و «آخرین» جداکننده. معمولاً گزینه‌ی Each Occurrence کاربرد بیشتری دارد.

- در پنجره‌ی Split to Column یک بخش به نام Advanced Options وجود دارد که تنظیمات مهمی دارد:
- Split into که انتخاب بین Columns یا Rows است، یعنی داده‌ها پس از تفکیک، بصورت ستون یا بصورت سطر قرار بگیرند.
 - Number of Columns to Split Into که تعیین می‌کند داده به چند ستون تقسیم شود. مثلاً اگر تاریخ به صورت روز/ماه/سال باشد، معمولاً ۳ بخش دارد. برای مثال اگر ما عدد ۲ را انتخاب کنیم، بخش سوم حذف می‌شود.
 - Quote Character که مشخص می‌کند آیا کاراکترهایی مثل " " (Double Quote) در نظر گرفته شوند یا خیر. مثلاً اگر داخل سلول "Hello, World" باشد، با فعال بودن این گزینه، ویرگول داخل کوتیشن‌ها نادیده گرفته می‌شود.
 - Insert Special Character که برای استفاده از کاراکترهای خاص مثل Tab، Carriage Return یا Line Feed که در جلسه‌ی اول و بخش Replace توضیح داده شد.

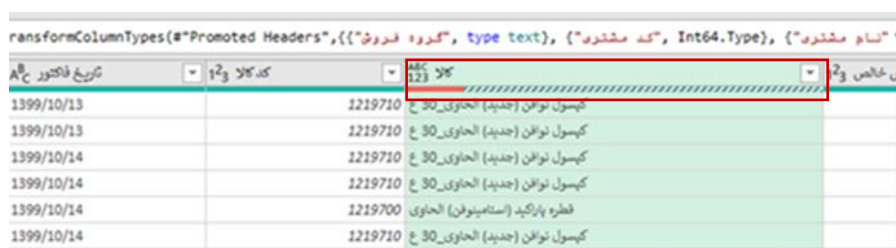
بعد از اجرای Power Query Split، به صورت خودکار یک مرحله جدید با نام Changed Type اضافه می کند. گاهی این مرحله را نمی خواهیم، چون ممکن است Power Query نوع داده را اشتباه تشخیص دهد. در این صورت می توانیم این مرحله را حذف کنیم.



دقت داشته باشیم که بعد از انجام Split، ستون های جدید ساخته می شوند و ستون اولیه (مثلاً تاریخ کامل) دیگر وجود ندارد و حذف می شود. برای اینکه آن ستون اصلی را نگه داریم، بهتر است قبل از Split با کلیک راست روی ستون، گزینه Duplicate Column را بزنیم تا یک نسخه از ستون اصلی حفظ شود. اگر مرحله ی Changed Type را حذف کنیم، داده های ستون های جدید به همان شکل متنی باقی می ماندند و Power Query به صورت خودکار نوع داده را تغییر نمی دهد. با این روش می توانیم داده ها را بسیار دقیق تر و تمیزتر تفکیک کنیم و کنترل بیشتری بر روی نحوه ی جدا شدن و نوع داده های خروجی داشته باشیم.

Remove Error/ Replace Error

در Power Query، وقتی فایلی را باز می کنیم، در همان نگاه اول می توانیم متوجه وجود خطاها (Errors) در داده ها شویم. در زیر ستون ها Column Profiling وجود دارد که نشان می دهد در هر ستون چه مقدار داده معتبر و یا خطا وجود دارد. اگر در ستون خطا وجود داشته باشد، بخش قرمز رنگی در خط زیر سرستون نمایان می شود که نشانگر سلول های خطا دار است. البته اگر حجم داده زیاد و تعداد خطا کم باشد، ممکن است بخش قرمز رنگ به آسانی دیده نشود، اما Power Query با دقت تمام آن را تشخیص داده و در Column Quality مشخص می کند. اگر داده ها کامل و بدون خطا بودند، رنگ نوار کاملاً سبز می شد، اما اگر بخشی از داده ها دارای خطا یا مقادیر نامعتبر باشند، Power Query رنگ نوار را به صورت هاشور زده (آبی و قرمز ترکیبی) نشان می دهد تا ما متوجه داده های ناخالص بشویم.



مشاهده‌ی خطاها در Power Query و اکسل

زمانی که داده‌ها از Power Query در اکسل، بارگذاری می‌شوند، اگر خطایی وجود داشته باشد، در شیت اکسل سلول‌های خطادار خالی نمایش داده می‌شوند، در حالی که در منبع اصلی، آن سلول‌ها دارای خطای مثلاً #N/A یا #VALUE! بوده‌اند. Power Query به‌صورت دقیق تعداد خطاها را گزارش می‌دهد و مشخص می‌کند در کدام ستون‌ها رخ داده‌اند و می‌توان آن سلول‌های خالی را با داده‌ای مناسب جایگزین کرد و یا اینکه آن سطرها را پاک کرد. (این بستگی به نیاز مخاطب دارد)

Table Name:

Properties External Table Data Table Style Options Table Styles

☐ Header Row ☐ First Column ☒ Filter Button

☐ Total Row ☐ Last Column

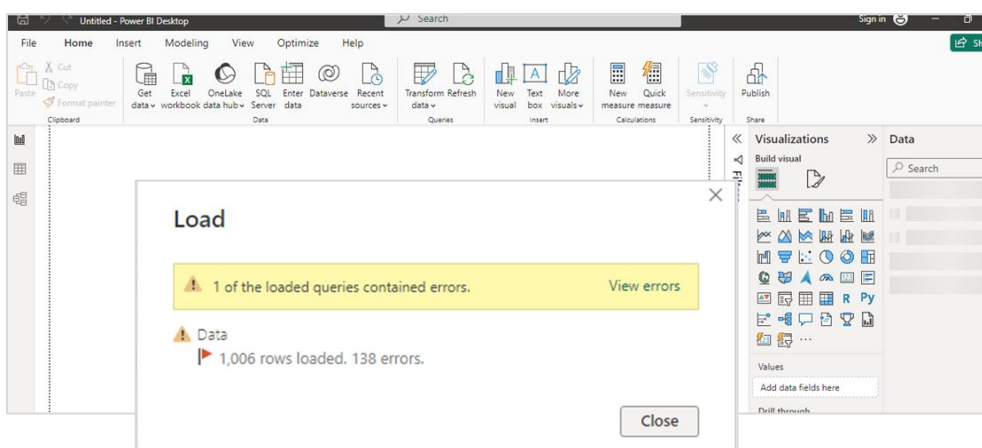
☒ Banded Rows ☐ Banded Columns

19

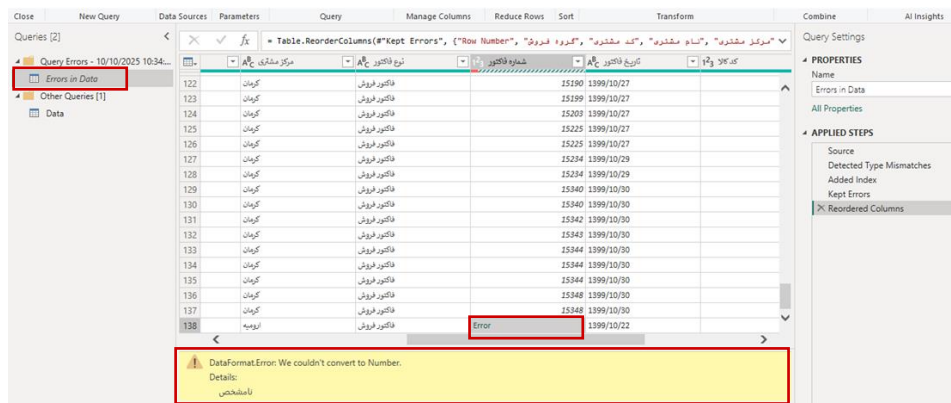
گروه فروش	نام مشتری	کد منطقه	نوع فاکتو	تاریخ فاکتو	کد کالا	میزان
گروه فروش زاهدان	دروخانه دکتر امیرپران	22267	فاکتور فروش	1399/10/21	1219801	6497
گروه فروش زاهدان	دروخانه دکتر امیرپران	22267	فاکتور فروش	1399/10/21	1219932	6502
گروه فروش زاهدان	دروخانه دکتر اسمعیل زالی	23548	فاکتور فروش	1399/10/21	1219725	6507
گروه فروش زاهدان	حیدری - پرواز - نعل - یسری	20736	فاکتور فروش	1399/10/13	1219904	6511
گروه فروش زاهدان	محمد علی دهقان نوری	24585	فاکتور فروش	1399/10/13	1219705	6514
گروه فروش زاهدان	محمد علی دهقان نوری	24585	فاکتور فروش	1399/10/21	1219706	6514
گروه فروش زاهدان	محمد علی دهقان نوری	24585	فاکتور فروش	1399/10/21	1219906	6514
گروه فروش زاهدان	دروخانه دکتر آذینسلو	21947	فاکتور فروش	1399/10/21	1219778	6519
گروه فروش زاهدان	دکتر برنی	26410	فاکتور فروش	1399/10/13	1219706	6526
گروه فروش زاهدان	دکتر برنی	26410	فاکتور فروش	1399/10/13	1219761	6526
گروه فروش زاهدان	دکتر برنی	26410	فاکتور فروش	1399/10/13	1219932	6526
گروه فروش زاهدان	دروخانه دکتر انصاری مقدم (امیر)	21874	فاکتور فروش	1399/10/21	1219801	6528
گروه فروش زاهدان	دروخانه دکتر انصاری مقدم (امیر)	21874	فاکتور فروش	1399/10/21	1219942	6528
گروه فروش زاهدان	دروخانه روزانه دکتر نقی زاده	27816	فاکتور فروش	1399/10/13	1219700	6530
گروه فروش زاهدان	دروخانه روزانه دکتر نقی زاده	27816	فاکتور فروش	1399/10/13	1219740	6530
گروه فروش زاهدان	دروخانه روزانه دکتر نقی زاده	27816	فاکتور فروش	1399/10/13	1219742	6530

138 errors.

در Power BI نیز هنگام بارگذاری داده‌ها، در زمان Load پیغام می‌دهد که چه تعداد سطر بارگذاری شده و چند سطر دارای خطا هستند. این همان اطلاعاتی است که در Power Query هم قابل مشاهده است.



در Power BI ، وقتی داده خطا دارد، در پنجره‌ی ورودی می‌توان روی گزینه‌ی View Errors کلیک کرد. با این کار، محیط Power Query باز می‌شود و کوئری جدیدی به نام Error in Data ساخته می‌شود که شامل تمام سطرهای خطادار است. در این صفحه می‌توانیم به راحتی خطاها را بررسی کنیم. هر خطا با جزئیات در پایین صفحه توضیح داده می‌شود:



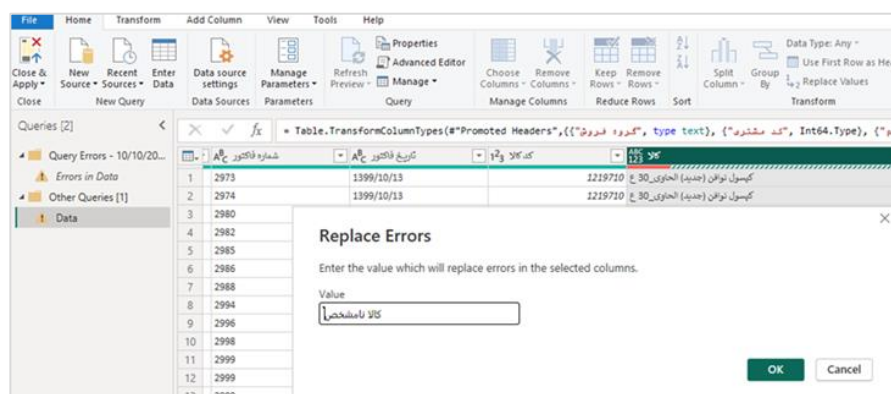
برای مثال، اگر در ستون «شماره فاکتور» مقداری متنی وجود داشته باشد ولی نوع داده ستون عدد (Number) تعریف شده باشد، Power Query نمی‌تواند آن مقدار را به عدد تبدیل کند و پیام خطا نمایش می‌دهد. یا در ستون «کالا»، ممکن است خطای #N/A وجود داشته باشد که به معنی مقدار نامشخص است.

گاهی در نمای اصلی جدول خطایی دیده نمی‌شود، اما در کوئری خطاها (Error Group) چندین خطا گزارش شده است. دلیل آن این است که Power Query برای سرعت بیشتر فقط ۱۰۰۰ سطر اول داده را در Column Profiling بررسی می‌کند. در نتیجه ممکن است خطاهایی در سطرهای بعدی وجود داشته باشند که در این بازه دیده نمی‌شوند. وقتی داده‌ها Load می‌شوند، Power Query خطاها را شناسایی کرده و در یک گروه جداگانه نمایش می‌دهد.

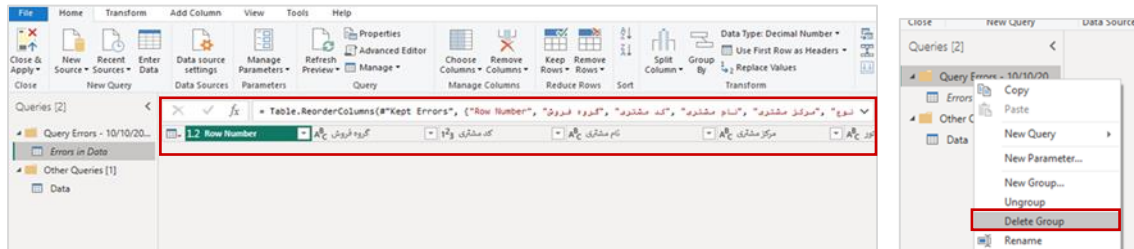
برای رفع خطاها، دو مسیر اصلی وجود دارد:

اگر بخواهیم تمام سطرهایی که دارای خطا هستند به‌طور کامل حذف شوند، روی ستون مورد نظر کلیک راست و گزینه **Remove Errors** را انتخاب کنید. با این کار کل سطرهایی که در آن ستون خطا دارند حذف می‌شوند. این روش زمانی مناسب است که خطاها اندک و غیرمهم باشند.

اگر بخواهیم خطاها حذف نشوند، بلکه جایگزین (Replace) شوند، روی ستون مورد نظر کلیک راست و گزینه **Replace Errors** را انتخاب کنید. سپس مقدار جایگزین را بنویسید، مثلاً "کالا نامشخص". این روش زمانی کاربرد دارد که بخواهیم ساختار داده حفظ شود و فقط مقدار خطاها تصحیح گردد.



زمانیکه خطاها برطرف شوند، جدول Error in ... خالی می‌شود. در این حالت می‌توانیم آن کوئری را حذف کنیم، روی کوئری Error in ... راست کلیک کنید. گزینه‌ی Delete Group را بزنید.



در نهایت، با انتخاب Close & Apply، داده‌ها بدون خطا در اکسل یا Power BI بارگذاری می‌شوند.

در اکسل نیز اگر خطایی وجود داشته باشد، در بخش Queries & Connections در سمت راست صفحه تعداد خطاها نشان داده می‌شود.

با کلیک روی آن، محیط Power Query باز می‌شود و کوئری Error in Data ساخته می‌شود. اما رفع خطا همیشه باید در کوئری اصلی انجام شود، درست مثل Power BI.

کد مشتری	نام مشتری	مرکز مشتری	نوع فاکتور	شماره فاکتور
1	شیرک بهداشت و درمان طبس	پورهند	فاکتور فروش	شماره فاکتور و...
2	داروخانه دکتر محمدپور	پورهند	فاکتور فروش	شماره فاکتور و...

اگر در Power Query به جای Load از گزینه‌ی Transform Data استفاده کنیم، می‌توانیم مستقیماً در همان محیط خطاها را بررسی و اصلاح کنیم، بدون اینکه ابتدا آن‌ها را در اکسل لود کنیم.

پس با استفاده از این دو ابزار Remove Errors و Replace Errors می‌توانیم داده‌ها را تمیز، پایدار و بدون خطا نگه داریم تا در مدل‌های داده‌ای و گزارش‌های تحلیلی خطایی رخ ندهد.

Rename Column

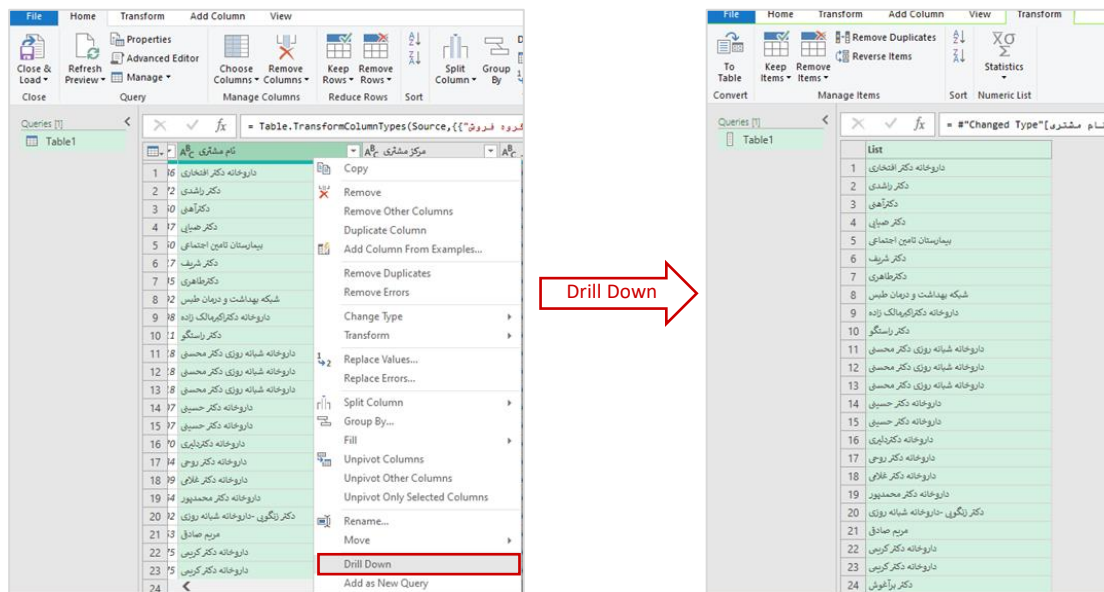
وقتی در Power Query نام یک ستون (Rename Column) را تغییر می‌دهیم، در بخش Applied Steps یک مرحله‌ی جدید به نام Renamed Columns ایجاد می‌شود. اما نکته‌ی جالب این است: اگر چند ستون را پشت‌سرهم و بدون انجام هیچ اقدام دیگری Rename کنیم، تمام آن تغییر نام‌ها در همان یک Step ذخیره می‌شوند. اما اگر بعد از Rename کردن، عملی دیگر (مثل حذف یا افزودن ستون) انجام دهیم و سپس دوباره یک ستون را Rename کنیم، Power Query آن را در Step جدید ذخیره می‌کند و با قبلی‌ها ادغام (Merge) نمی‌شود.

نوع فاکتور	Center	Customer	کد مشتری
فاکتور فروش	پورهند	داروخانه دکتر الهادی	25886
فاکتور فروش	پورهند	دکتر رابندی	22072
فاکتور فروش	پورهند	دکتر آری	8050
فاکتور فروش	پورهند	دکتر حبیبی	8047
فاکتور فروش	پورهند	بیمارستان تامین اجتماعی	19850
فاکتور فروش	پورهند	دکتر شریف	8017
فاکتور فروش	پورهند	دکتر طاهری	8045
فاکتور فروش	پورهند	شیرک بهداشت و درمان طبس	26392
فاکتور فروش	پورهند	داروخانه دکتر کریم‌پور	23498
فاکتور فروش	پورهند	دکتر راسنگی	21311

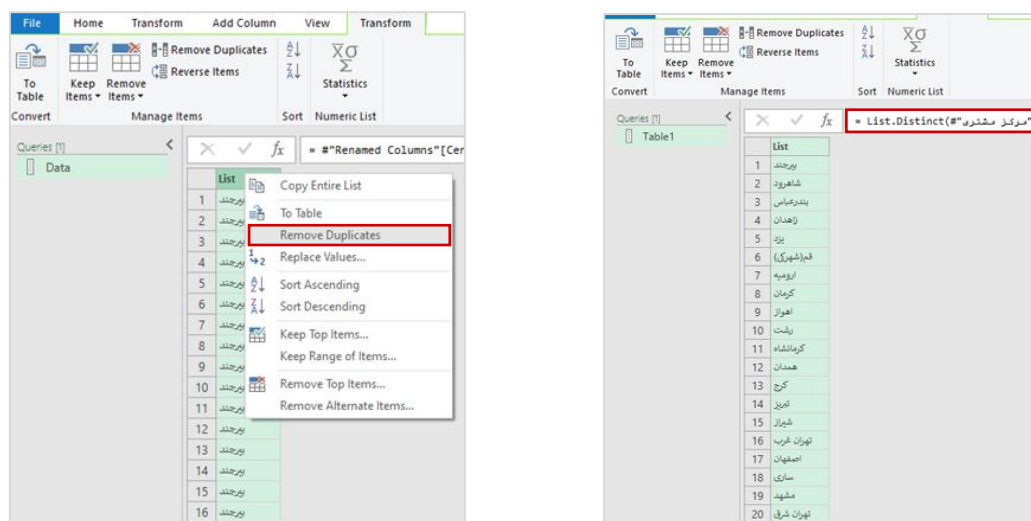
در واقع، بعضی از اکشن‌ها در Power Query قابلیت ادغام تغییرات (Combine Operations) دارند، مثل Rename Columns. در این حالت Power Query این تغییرات را به صورت یک SubList در فرمول زبان M ذخیره می‌کند. (به کد نوشته شده در قسمت Formula Bar دقت کنید.)

Drill Down

لیست‌ها (Lists) یکی از پایه‌ای‌ترین ساختارهای داده در زبان M هستند. در واقع، هر نوع داده‌ای مثل آرایه‌ها یا ستون‌های اکسل در Power Query نیز در قالب لیست شناخته می‌شوند. برای تبدیل یک ستون به لیست، روی ستون مورد نظر راست کلیک کنید، گزینه‌ی Drill Down را بزنید. با این کار، Power Query فقط همان ستون را نگه می‌دارد و کل جدول به یک List تبدیل می‌شود. هر سلول از آن ستون، یک عضو از لیست خواهد بود.



اکنون می‌توانیم روی این لیست کارهای مختلفی انجام دهیم، برای مثال Remove Duplicate که مقادیر تکراری را حذف می‌کند که تابع List.Distinct می‌شود. این لیست‌ها در زبان M بسیار پرکاربرد هستند، چون می‌توان آن‌ها را در فرمول‌ها، فیلترها، پارامترها و حتی ساخت جداول جدید به کار برد.



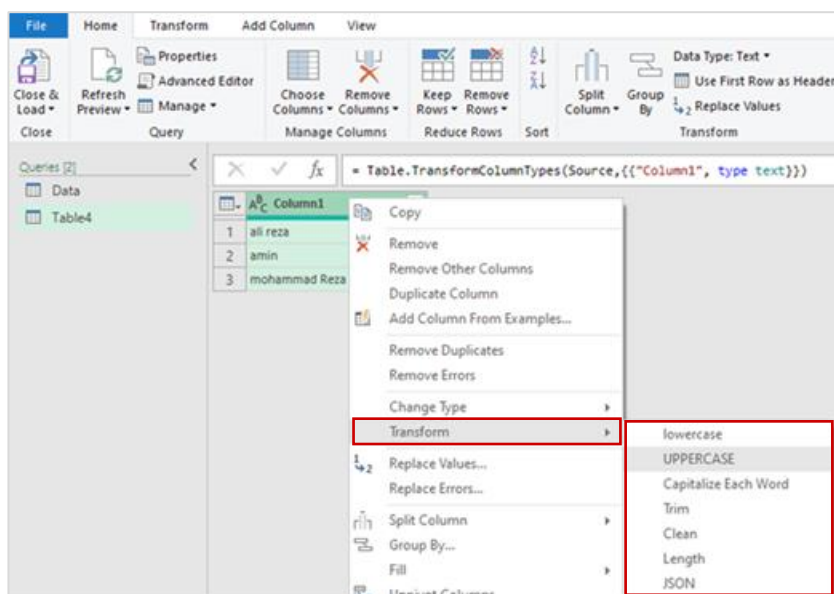
Transform

بخش Transform در Power Query مجموعه‌ای از دستورات پرکاربرد برای تغییر محتوای داده‌ها است.

این گزینه‌ها شامل موارد زیر هستند:

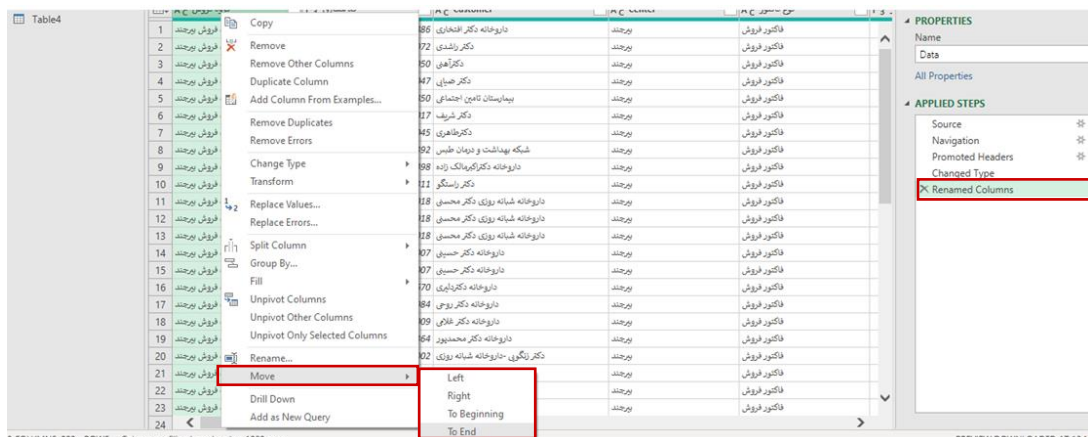
- Trim: حذف فاصله‌های اضافی از ابتدا و انتهای متن
- Clean: حذف کاراکترهای غیرقابل چاپ
- Upper Case: تبدیل تمام حروف به حروف بزرگ
- Lower Case: تبدیل حروف به حروف کوچک
- Capitalize Each Word: بزرگ کردن حرف اول هر کلمه
- Length: محاسبه‌ی تعداد کاراکترهای هر سلول

هر یک از این اقدامات، یک Step جدید در Applied Steps ایجاد می‌کند و می‌تواند برای پاک‌سازی یا استانداردسازی داده‌ها بسیار مؤثر باشد.



Move Column

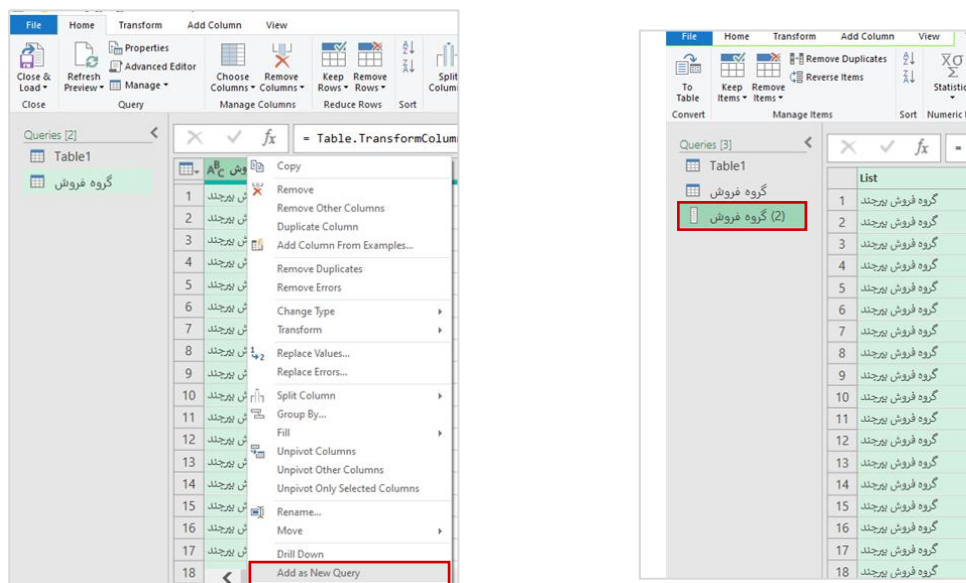
در Power Query می‌توان با (Drag & Drop) ستون‌ها را جابه‌جا کرد، اما علاوه بر آن، گزینه‌ای به نام Move هم وجود دارد که برای جابه‌جایی دقیق‌تر استفاده می‌شود. با کلیک راست روی ستون و انتخاب Move، چهار گزینه نمایش داده می‌شود:



این قابلیت زمانی مفید است که جدول تعداد زیادی ستون داشته باشد و بخواهیم ترتیب نمایش آن‌ها را تنظیم کنیم. بعد از اجرای این عملیات، یک Step جدید با نام Reordered Columns در Applied Steps اضافه می‌شود.

Add As New Query

گزینه‌ی New Query در Power Query امکان ایجاد یک کوئری تازه از داده‌های موجود را می‌دهد. این عمل مشابه Drill Down است، با این تفاوت که در Drill Down، خود جدول اصلی به لیست یا بخش خاصی محدود می‌شود. در New Query، یک نسخه‌ی جدید (Copy) از جدول ساخته می‌شود که مستقل از کوئری اصلی عمل می‌کند.



این کوئری جدید می‌تواند جداگانه Refresh شود و تنظیمات مخصوص خودش را داشته باشد. به عنوان مثال، می‌توان از جدول اصلی یک کوئری جدید ساخت برای استخراج فقط داده‌های خاص یا انجام پردازش جداگانه بدون تغییر در جدول اصلی.

باز هم توجه به این نکته لازم است که Power Query تمام این تغییرات را در قالب Steps ذخیره می‌کند، و هر Step در واقع یک خط از زبان M است. با کلیک روی آیکون fx در بالای پنجره، می‌توانیم مستقیماً فرمول هر مرحله را ویرایش یا شخصی‌سازی کنیم.

جلسه سوم

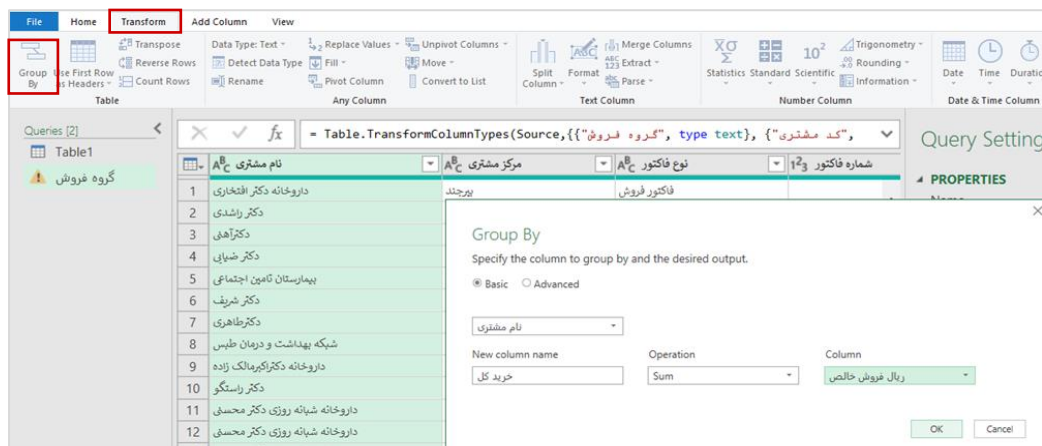
خیلی از اوقات داده‌هایی که از اکسل یا صفحه‌گسترده‌ها می‌گیریم، طراحی درستی ندارند و نیاز به تغییر شکل (Transform) در آن‌ها وجود دارد تا بتوانیم تحلیل درستی روی داده انجام دهیم و گزارش بگیریم. در این جلسه دستورات کامل Transform آموزش داده می‌شود.

Transform

Group By

با Power Query می‌توانیم بعضی کارهای از جنس تحلیل و گزارش را انجام دهیم، اما این ابزار بیشتر برای پاک‌سازی داده است، نه تحلیل. گزینه Group By همان کاری را انجام می‌دهد که در اکسل با Pivot Table انجام می‌دهیم یا در SQL با دستور Group By در Statement.

برای مثال، فرض کنید می‌خواهیم از مشتریان گزارشی بگیریم که مجموع ریال خریدهای آن‌ها را بدانیم. در تب Transform و گزینه Group By با انتخاب گزینه Basic، نام ستون مورد نظر را داریم، نام ستون جدید و عملیاتی که می‌خواهیم انجام دهیم (مثلاً Sum، Average یا هر چیز دیگر) و ستونی که می‌خواهیم مبنای محاسبه باشد (مثلاً ستون «ریال خریده‌ها» در این حالت، به ازای هر مشتری، جمع کل ریالی خرید او را خواهیم داشت. در واقع می‌خواهیم مشتریان را به صورت Unique در یک ستون داشته باشیم و در ستون مقابل، جمع خریدهایشان را مشخص کنیم.



در حالت Advanced، امکانات بیشتری وجود دارد؛ یعنی می‌توانیم نام ستون مورد نظر را به‌علاوه یک ستون دیگر هم در نظر بگیریم. مثلاً ستون اول «مشتریان» است، ستون دوم «کالاها» و می‌توانیم چند ستون جدید هم تعریف کنیم و عملیات مختلفی روی آن‌ها انجام دهیم. در این صورت، یک Step اضافی به نام Sorted Rows اضافه می‌شود.

Group By
Specify the columns to group by and one or more outputs.

☐ Basic ☒ Advanced

نام مشتری
ایلا

Add grouping

New column name	Operation	Column
خرید کل	Sum	ریال فروش خالص
تعداد فاکتور	Count Rows	

Add aggregation

OK Cancel

Table1

گروه فروش

نام مشتری	ایلا	خرید کل	تعداد فاکتور
داروخانه دکتر احمدی	کیسول توان (عمید) 30 ع	1045050	1
دکتر راشدی	کیسول توان (عمید) 30 ع	2090100	1
دکتر افشار	کیسول توان (عمید) 30 ع	1045050	1
دکتر صافی	کیسول توان (عمید) 30 ع	4180200	1
بیمارستان کاین احمدی	قطره پاراکتید (اسامیون) الحاقی	3813550	1
دکتر شریف	کیسول توان (عمید) 30 ع	1463070	2
دکتر قاضی	سوسپانسیون تری دی تری (کسیم+میزیم+ویتامین D3) 120 م ل الحاقی	833800	1
شبکه بهداشت و درمان طیس	کیسول مولی ویتامین + میرال الحاقی 100 ع	320016000	1
داروخانه دکتر کارمیرانک زاده	کیسول توان (عمید) 30 ع	6270300	2
دکتر راستگو	کیسول توان (عمید) 30 ع	4180200	1
داروخانه شبانه روزی دکتر محسن	سوسپانسیون تری دی تری (کسیم+میزیم+ویتامین D3) 120 م ل الحاقی	833800	1
دکتر قاضی	سوسپانسیون تری دی تری (کسیم+میزیم+ویتامین D3) 120 م ل الحاقی	833800	1
خرید کل		833800	1
تعداد فاکتور			2

Query Settings

PROPERTIES

Name

گروه فروش

APPLIED STEPS

Source

Changed Type

Grouped Rows

همچنان توجه داشته باشیم که در Power Query معمولاً از دستور Group By برای گزارش گیری استفاده نمی شود؛ بلکه در فرایند پاک سازی داده گاهی لازم است تغییراتی انجام شود که Group By یکی از آنهاست. مثلاً گاهی لازم است مقادیر را به هم چسبانده شوند (Concatenate) که در اینجا Group By به کار می آید، اما صرفاً برای گزارش گرفتن نیست، بلکه یکی از Step های پاک سازی داده محسوب می شود و تغییر شکلی در جدول ایجاد می کند که به آن نیاز داریم.

Use First Row as Headers

این گزینه شامل دو بخش است:

- Use First Row as Headers
- Use Headers as First Row

File Home Transform Add Column View

Group By

Use First Row as Headers

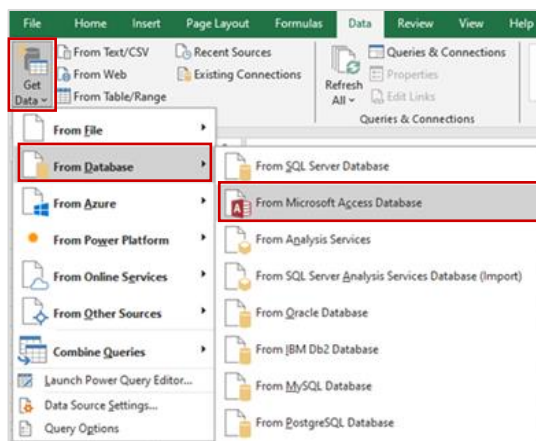
Use Headers as First Row

Promote the first row of this table into column headers.

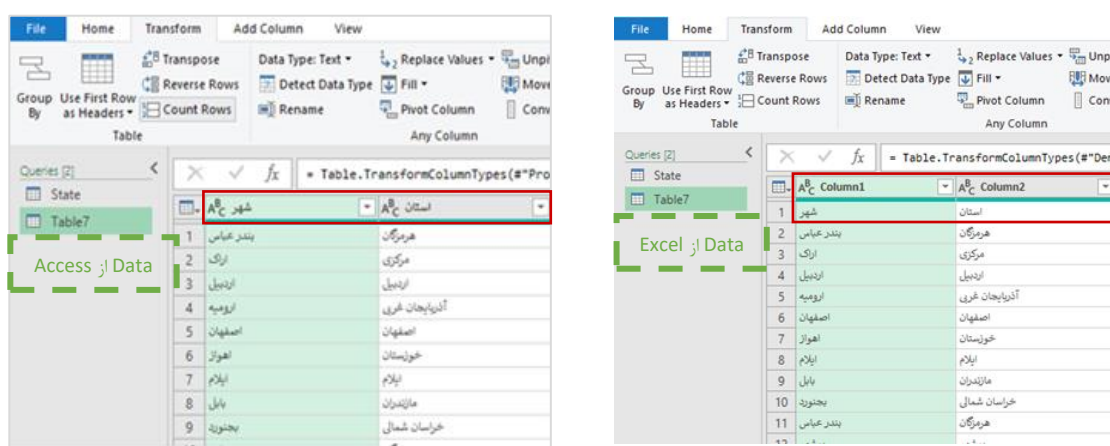
Columns

1	2	3
استان	شهر	شماره
تهران	تهران	1
تهران	تهران	2
تهران	تهران	3
تهران	تهران	4
تهران	تهران	5
تهران	تهران	6
تهران	تهران	7
تهران	تهران	8

اگر از فایل Access داده‌ای را Import کنید، Power Query می‌تواند Table و Query را بخواند. اما اگر Query پارامتریک در Access داشته باشید (یعنی نیاز به ورود مقدار باشد)، آن Query در Power Query خوانده نمی‌شود. بدلیل آنکه آن پارامتر در Access تعریف شده و در Power Query قابل خواندن نیست.

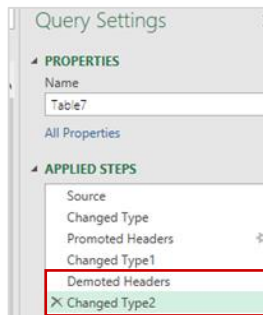


جدولی که از Access در Power Query آورده می‌شود، سرستون‌ها (Headers) به‌درستی تشخیص داده می‌شود، زیرا Access، SQL و سایر Database ها ساختارمند هستند و همه آن‌ها دارای جدول و هدر هستند. اما در اکسل این‌طور نیست و تمام داده‌ها در Sheet و Cell ها وارد می‌شوند و ممکن است ساختار خاصی برای آن‌ها تعریف نشده باشد. گاهی اوقات به دلیل متنی بودن همه داده‌ها (چه سرستون‌ها و چه خود داده‌ها)، Power Query سطر اول را به عنوان سرستون (Header) تشخیص نمی‌دهد و آن را مانند یک سطر عادی تلقی می‌کند. اما گزینه‌ای وجود دارد که بر اساس نیاز، می‌توان سطر اول را به سرستون منتقل کرد؛ این گزینه Use First Row as Headers نام دارد.



گاهی نیز برعکس آن را نیاز داریم؛ یعنی نمی‌خواهیم سرستون‌ها در بالا باشند، بلکه می‌خواهیم آن‌ها را به عنوان سطر معمولی در جدول قرار دهیم. در این حالت، از گزینه Use Headers as First Row استفاده می‌کنیم.

با اجرای این دستور، دو Step جدید به نامهای Promoted Header و Changed Type اضافه می‌شود و نوع داده‌ها را به صورت خودکار تشخیص می‌دهد.



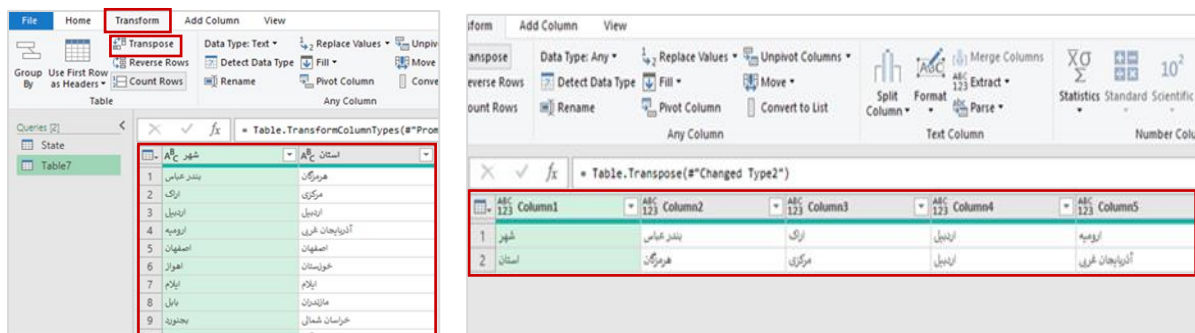
Transpose

به معنای تغییر جای سطرها و ستون‌ها است. در ریاضیات، مشابه این عمل را در مبحث ماتریس‌ها داریم که به آن ترانپو (Transpose) گفته می‌شود.

Transpose of a Matrix

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3} \quad A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}_{3 \times 2}$$

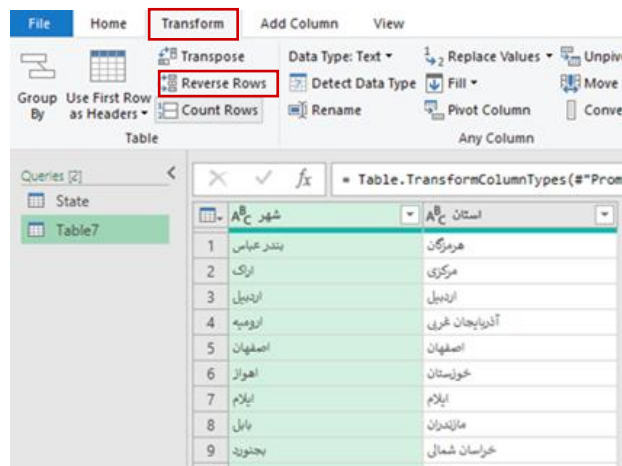
در این تصویر یک ماتریس دو ردیف در سه ستون، به ماتریس سه ردیف در دو ستون تبدیل شده است.



در جلسات بعدی که سناریوهای کاربردی بررسی می‌شوند، از Transpose در یکی از مثال‌ها استفاده می‌کنیم تا کاربرد آن مشخص شود.

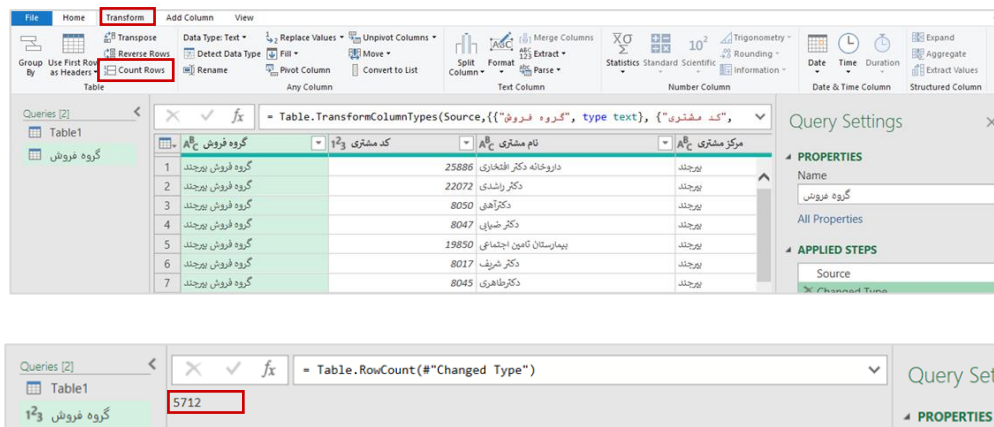
Reverse Rows

این دستور داده‌ها را برعکس می‌کند و جدول را از پایین به بالا می‌چیند. البته این عمل با Sort کردن متفاوت است. به دلیل ویژگی Lazy Load در Power Query، برای جدول‌هایی با سطرهای زیاد که سطرهای پایینی دیده نمی‌شوند، می‌توانیم با اجرای Reverse Rows سطرهای پایینی را مشاهده کرده و موضوع مورد نظر را بررسی کنیم.



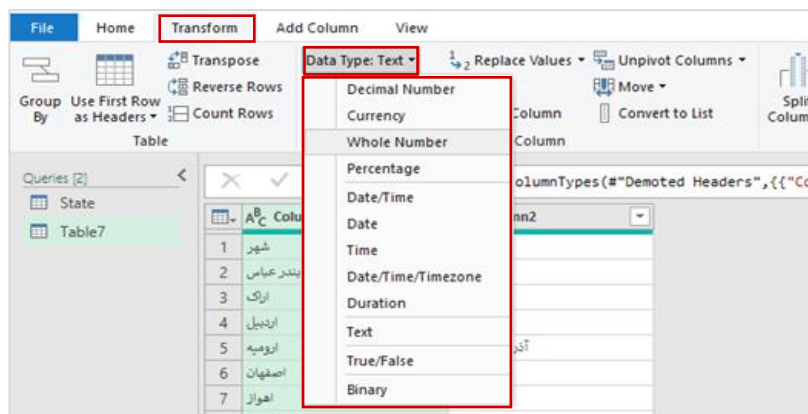
Count Rows

این دستور جدول را جمع می‌کند و فقط تعداد سطرهای موجود در جدول را نمایش می‌دهد.



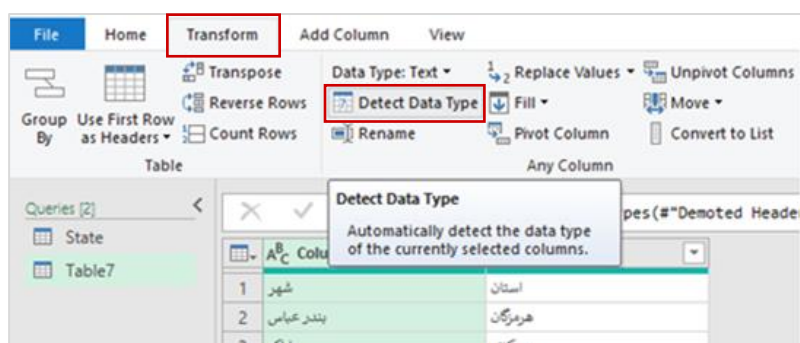
Data Type

این همان قابلیت است که با کلیک روی آیکون کوچک کنار نام ستون (علامت ABC یا 123) می‌توان نوع داده را مشخص کرد.



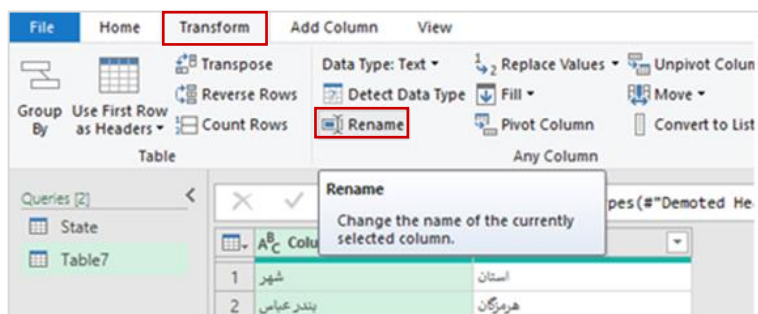
Detect Data Type

در درس اول توضیح داده شد که به صورت خودکار نوع داده‌ها را تشخیص می‌دهد.



Rename

می‌توان نام ستون را تغییر داد که پیشتر در درس اول توضیح داده شد.

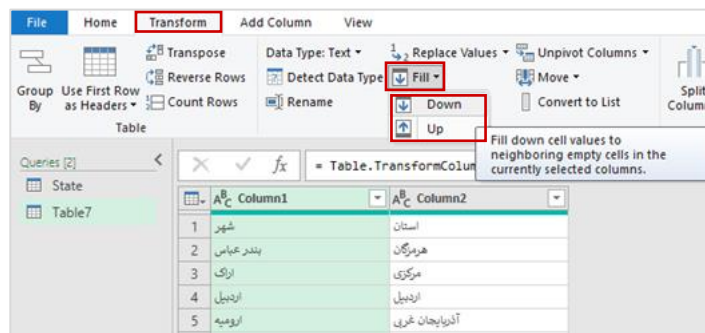


Replace Value

جایگزینی مقادیر که در جلسه‌ی دوم آموزش داده شد.

Fill Down / Fill Up

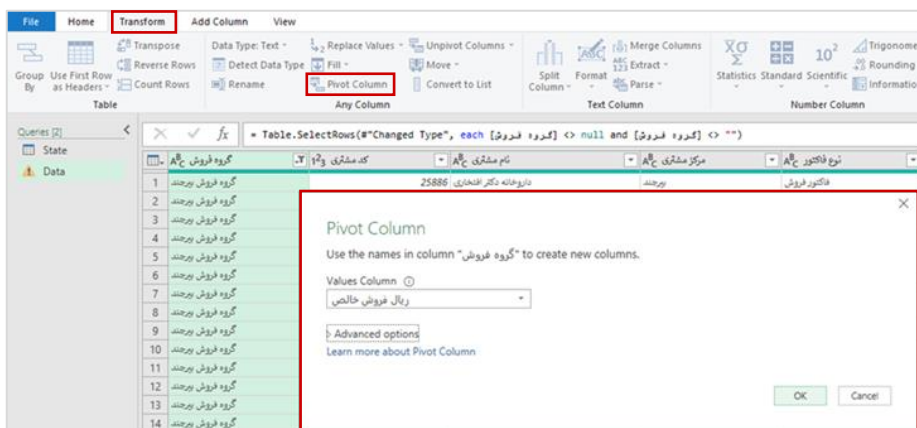
در جلسات قبل معرفی شدند. در جلسات آینده، به‌ویژه در سناریوهای کاربردی، از آن‌ها به شکل گسترده‌تر استفاده می‌گردد.



Pivot Column

Pivot در واقع به معنای «چرخاندن» است، در دنیای داده نیز، Pivot Table یا Pivot Column به نوعی از جدول گفته می‌شود که در آن، سرستون‌ها (Headers) به جای اینکه شامل نام‌های کلی باشند، در واقع مقادیر (Values) هستند. به عبارت دیگر، جدول Pivot حالتی غیر استاندارد از داده است.

در جدول Group By شده هم می‌توان عملیات Pivot را انجام داد، به عنوان مثال، اگر ستون "گروه مشتری" را انتخاب و سپس گزینه‌ی Pivot Column را کلیک کنیم، در قسمت Values Column (ستون مقادیر)، مشخص کنیم که مقدار مورد نظر "ریال فروش خالص" باشد. به این ترتیب، داده‌ها بر اساس گروه فروش چرخیده و هر گروه به عنوان یک ستون جدید نمایش داده می‌شود.



به این ترتیب، انگار جدولی که در بالا داشتیم (فرم استاندارد) را به جدول پایینی (فرم چرخیده یا Pivot شده) تبدیل کرده‌ایم. یعنی جدول را از حالت استاندارد خارج کرده‌ایم.

دلیل انجام این کار این است که در برخی مراحل از فرآیند پاک‌سازی داده‌ها، لازم است موقتاً داده‌ها را به حالت Pivot دربیآوریم تا بتوانیم تغییرات خاصی روی ساختارشان انجام دهیم. اگر بخواهیم جدول را دوباره به حالت اولیه (استاندارد) برگردانیم، نیازی به استفاده از دستور Unpivot نیست، بلکه کافی است Step مربوط به Pivot را حذف کنیم تا داده‌ها به حالت قبل برگردند.

Unpivot Columns

این دستور زمانی استفاده می‌شود که بخواهیم جدول را به حالت استاندارد برگردانیم. گاهی داده‌ای که به ما داده شده، در حالت استاندارد نیست (مثلاً مقادیر به جای سطر، در ستون‌ها پخش شده‌اند). در این صورت، باید با دستور Unpivot آن را به حالت درست دربیآوریم.

برای مثال، فرض کنید جدولی به نام Target داریم که سرستون‌های آن شامل مقادیر ماه و مقدار فروش هستند. اگر تعداد ستون‌هایی که باید Unpivot شوند زیاد باشد، کافی است روی یکی از ستون‌ها کلیک کنیم، و از زبانه Transform، گزینه Unpivot Other Columns را انتخاب کنیم تا تمام ستون‌های دیگر Unpivot شوند و جدول به حالت استاندارد دربیاید.

در این حالت، سرستون‌ها معمولاً تبدیل به دو ستون جدید به نام‌های Attribute و Value می‌شوند. می‌توان نام آن‌ها را تغییر داد.

Column1	Attribute	Value
گسترش داروی ایران	فروردین	2000
گسترش داروی ایران	اردیبهشت	1500
گسترش داروی ایران	خرداد	8000
گسترش داروی ایران	تیر	400
گسترش داروی ایران	مرداد	1000
گسترش داروی ایران	شهریور	1000
گسترش داروی ایران	مهر	3000
گسترش داروی ایران	آبان	6000
گسترش داروی ایران	آذر	4000
گسترش داروی ایران	دی	3000
گسترش داروی ایران	بهمن	2000

باید توجه داشت که جدول در این حالت به شکل استاندارد (ستونی و قابل تحلیل) درآمده است. اما گاهی در فرآیند پاک‌سازی لازم است دوباره آن را به حالت Pivot برگردانیم، که در آن صورت از Pivot Column استفاده می‌کنیم.

Move

این گزینه در کلیک راست روی ستون‌ها نیز وجود دارد. این گزینه به ما امکان می‌دهد ستون را به راست، چپ، ابتدا یا انتهای جدول منتقل کنیم. (در جلسه قبل نیز توضیح داده شد).

Convert to List

این دستور جدول را به List تبدیل می‌کند، یعنی تمام مقادیر ستون انتخاب‌شده را به صورت فهرستی از مقادیر نمایش

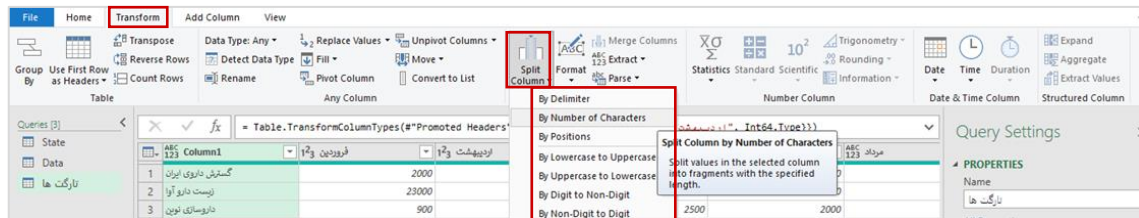
می‌دهد.

گروه فروش	کد مشتری	نام مشتری
گروه فروش بیرجند	25886	داروخانه دکتر افتخاری
گروه فروش شاهرود	26813	داروخانه دکتر عسکری
پندرعباس	20977	داروخانه دکتر محمدی
گروه فروش زاهدان	24587	دکتر عصمت حجازی زاده
گروه فروش یزد	20736	حیدری (یزد-نعل اسبی)
گروه فروش قم	22597	داروخانه سیدالشهداء-تفرش

گروه فروش	List
گروه فروش بیرجند	1
گروه فروش شاهرود	2
پندرعباس	3
گروه فروش زاهدان	4
گروه فروش یزد	5

Split Column

در جلسات قبل توضیح داده شد که با استفاده از آن می‌توانیم یک ستون را بر اساس جداکننده‌ها (Delimiter) یا تعداد کاراکترها تقسیم کنیم.

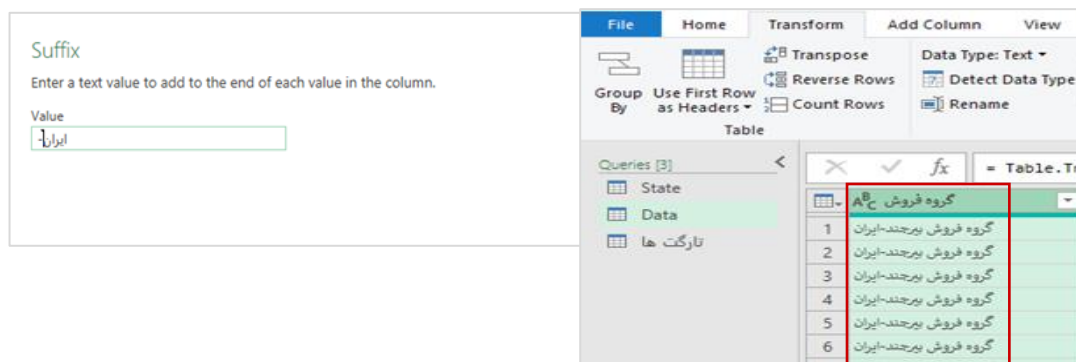
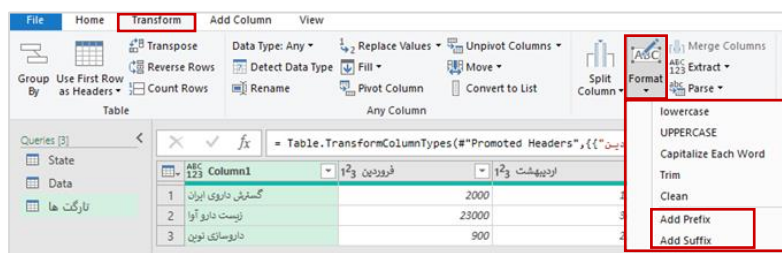


Format

برخی از گزینه‌های آن در کلیک راست Transform نیز وجود دارد (مانند Trim، Clean، Uppercase و...)، اما دو گزینه‌ی خاص هم در این بخش وجود دارد:

۱. Add Prefix : اضافه کردن پیشوند به داده‌های یک ستون

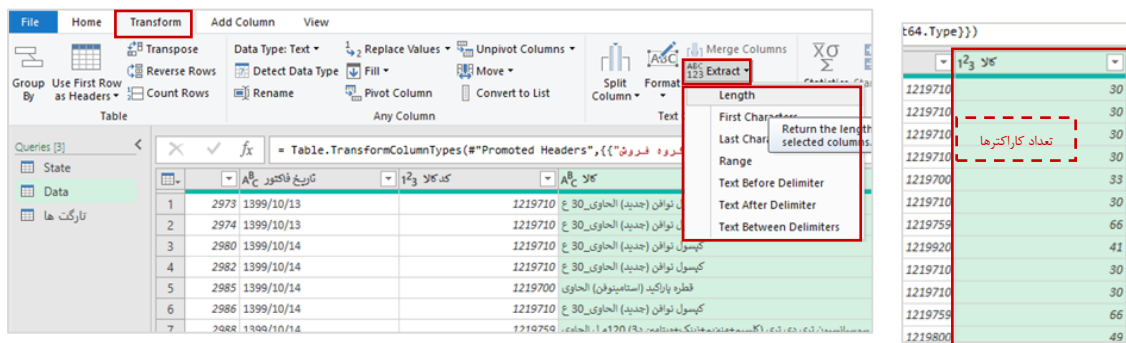
۲. Add Suffix : اضافه کردن پسوند به داده‌های یک ستون



Extract

این گزینه برای جدا کردن بخشی از متن در یک ستون متنی (Text Column) است. گزینه‌های مهم آن عبارت‌اند از:

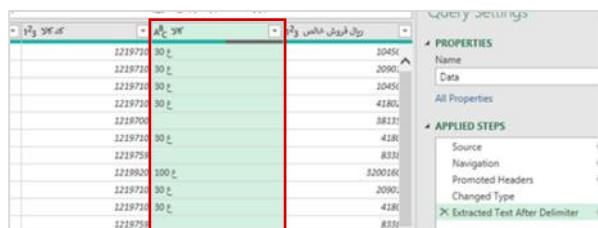
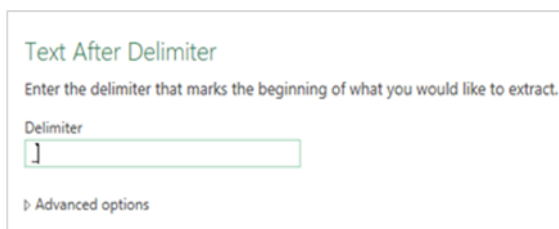
- Length : تعداد کاراکترهای هر سلول (مثلاً در ستون شماره، نشان می‌دهد هر مقدار چند کاراکتر دارد)



- First Characters : نمایش چند کاراکتر اول
- Last Characters : نمایش چند کاراکتر آخر
- Range : مشخص می‌کند از کاراکتر چندم تا چندم را جدا کند
- Text Before Delimiter : جدا کردن متن قبل از جداکننده
- Text After Delimiter : جدا کردن متن بعد از جداکننده
- Text Between Delimiters : جدا کردن متن بین دو جداکننده

برای مثال، اگر در داده‌ها علامت Underline (_) وجود داشته باشد، می‌توانیم با گزینه‌ی Text After Delimiter

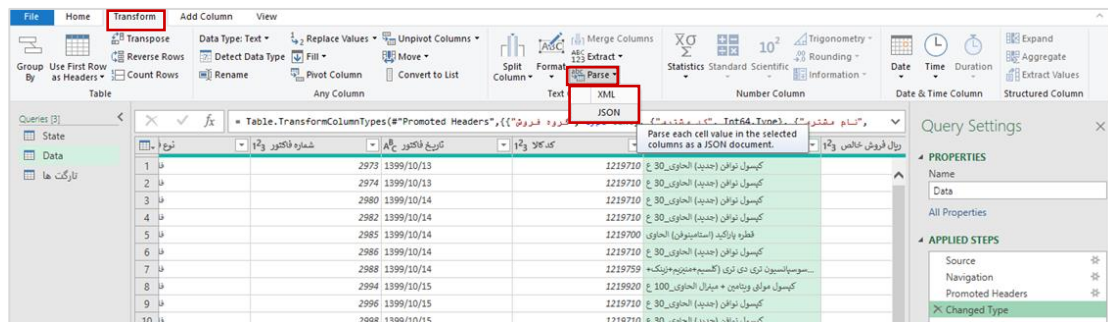
مشخص کنیم که هر چیزی بعد از Underline نمایش داده شود.



Parse

گاهی داده‌هایی از جنس JSON یا XML داریم. با استفاده از گزینه‌ی Parse، می‌توان این نوع داده‌ها را خواند و به

ساختار جدولی (Table) تبدیل کرد. البته این مباحث تخصصی‌تر و پیشرفته‌تر هستند و در سطح مقدماتی به آن‌ها پرداخته نمی‌شود.



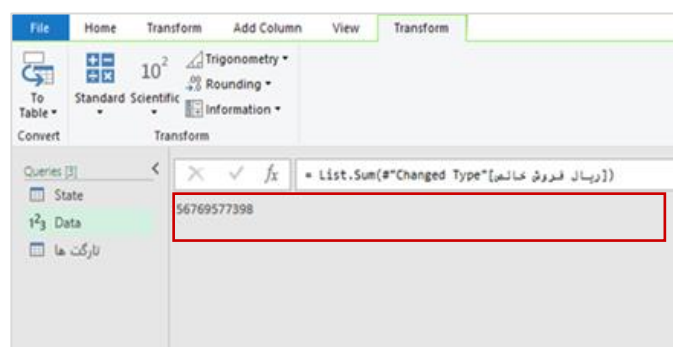
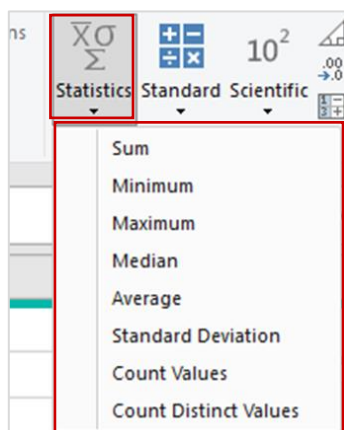
Statistics

این گزینه مخصوص ستون‌های عددی است و با آن می‌توان محاسبات آماری مختلفی انجام داد، مانند:

- Sum (جمع ستون)
- Maximum (بیشترین مقدار)
- Minimum (کمترین مقدار)
- Average (میانگین)
- Standard Deviation (انحراف معیار)

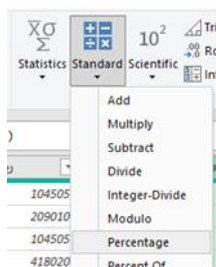
و سایر توابع آماری مشابه.

برای مثال با انتخاب گزینه Sum، نتیجه جمع یک ستون اینطور ظاهر می‌شود:



Standard

برای انجام محاسبات ریاضی پایه روی ستون‌ها استفاده می‌شود. گزینه‌های مهم آن عبارت‌اند از:

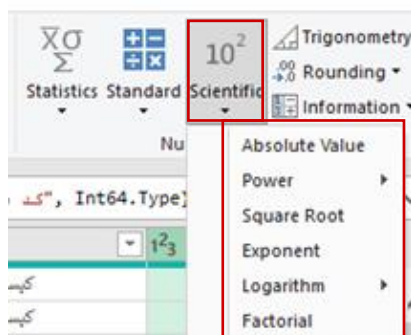


- Add : جمع با عدد مشخص
- Subtract : تفریق
- Multiply : ضرب
- Divide : تقسیم
- Integer Divide : تقسیم صحیح (عدد اعشاری را حذف می‌کند)
- Percentage : محاسبه درصدی از عدد (مثلاً ۲٪ از هر مقدار)
- Modulo : نمایش باقیمانده‌ی تقسیم (Remainder)

به‌عنوان مثال، اگر بخواهیم بدانیم ۲ درصد هر مقدار چقدر است، می‌توانیم از Percentage استفاده کنیم و نتیجه مستقیماً در همان ستون یا ستون جدید نمایش داده می‌شود.

Scientific

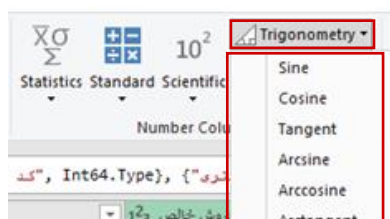
شامل توابع ریاضی پیشرفته‌تر می‌شود، از جمله:



- Absolute Value (قدر مطلق)
- Power (توان)
- Square Root (جذر)
- و سایر محاسبات علمی مشابه.

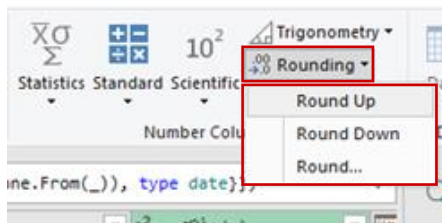
Trigonometry

در این بخش توابع مثلثاتی مانند Sin, Cos, Tan (سینوس، کسینوس، تانژانت) در دسترس هستند.



Rounding

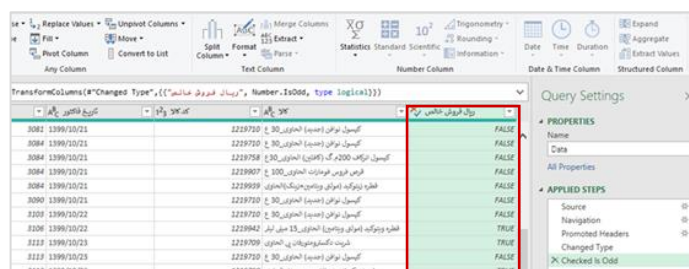
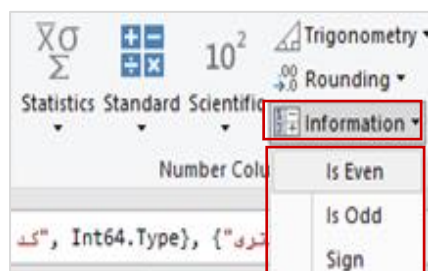
در این قسمت گزینه‌هایی برای تنظیم تعداد اعشار یا گرد کردن مقادیر عددی استفاده می‌شود.



- Round : گرد کردن معمولی
- Round Up : گرد کردن به بالا
- Round Down : گرد کردن به پایین

Information

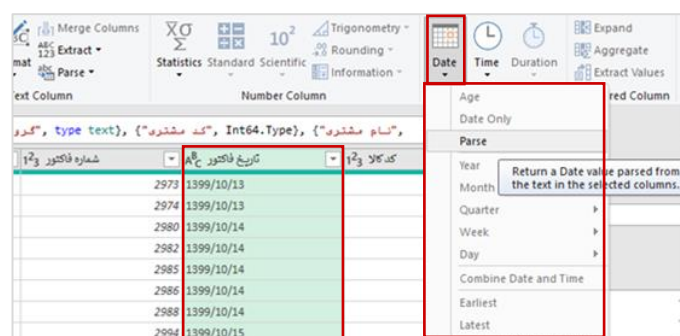
این بخش برای بررسی ویژگی‌های داده‌هاست. به‌عنوان مثال، اگر عدد فرد یا زوج باشد، مقدار True یا False را برمی‌گرداند. همچنین می‌تواند تشخیص دهد که آیا سلول خالی است یا خیر، یا نوع داده را بررسی کند.

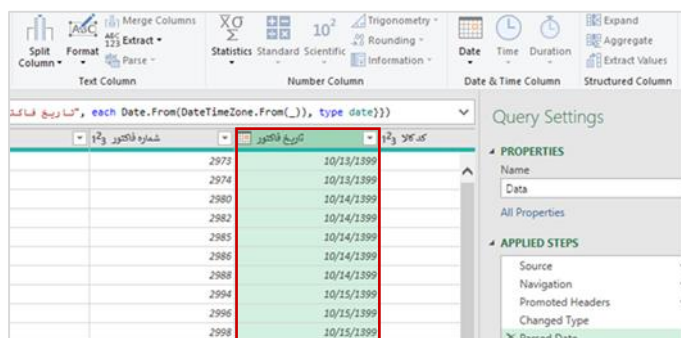


Date & Time

در این قسمت می‌توان عملیات مختلفی روی تاریخ و زمان انجام داد. گزینه‌ی Parse در این بخش زمانی استفاده

می‌شود که بخواهیم نوع داده را از Text به Date یا Time تبدیل کنیم. به‌طور مشابه، می‌توان تاریخ را به زمان، یا زمان را به تاریخ تغییر داد.





Create Data Type

در واقع این ابزار جزو Transformation نیست، بلکه برای ایجاد نوع داده سفارشی (Custom Data Type) در اکسل استفاده می‌شود. به عنوان مثال، اگر جدولی به نام Target داشته باشیم، با انتخاب گزینه‌ی Data Type و رفتن به بخش Advanced، می‌توانیم تعیین کنیم که نام Data Type مثلاً «ماهانه» (Monthly) باشد و Display Column مثلاً ستون «ماه» (Month) باشد. سپس با زدن Add و OK، این نوع داده در اکسل ایجاد می‌شود و می‌توان روی آن تحلیل و محاسبات انجام داد. اما توجه کنید که Power Query در Power BI، چنین گزینه‌ای وجود ندارد و مخصوص اکسل است.

Merge & Append

دستورات این درس در بخش Combine شامل Append و Merge است. در واقع، اضافه کردن دیتاها در سطرها معنی دارد و حفظ شدن سطرها مهم باشد. مثلاً زمانی که سفارشات جدید می‌آیند، باید به سطرها اضافه شوند (اگر جدول استاندارد باشد). اگر شعبات جدید اضافه شوند، از دستور Append استفاده می‌کنیم.

دستور Append

در Append کردن، هم‌نام بودن ستون‌ها مهم است، اما ترتیب آن‌ها اهمیتی ندارد. مثلاً در یک جدول در سرستون نوشته شده "A2" و در جدول دیگر هم در سرستون نوشته شده "A 2" (یک فاصله میان دو کاراکتر وجود دارد)، در واقع این دو با هم فرق دارند و دو ستون محسوب می‌شوند، که با Append کردن، اطلاعات هر دو ستون در جدول جدید آورده می‌شود و ترتیب آمدن ستون‌ها براساس انتخاب اولین جدول است که جدول A آورده شود یا جدول B.

ستون 1	ستون 2

ستون 1	ستون 2	ستون 3

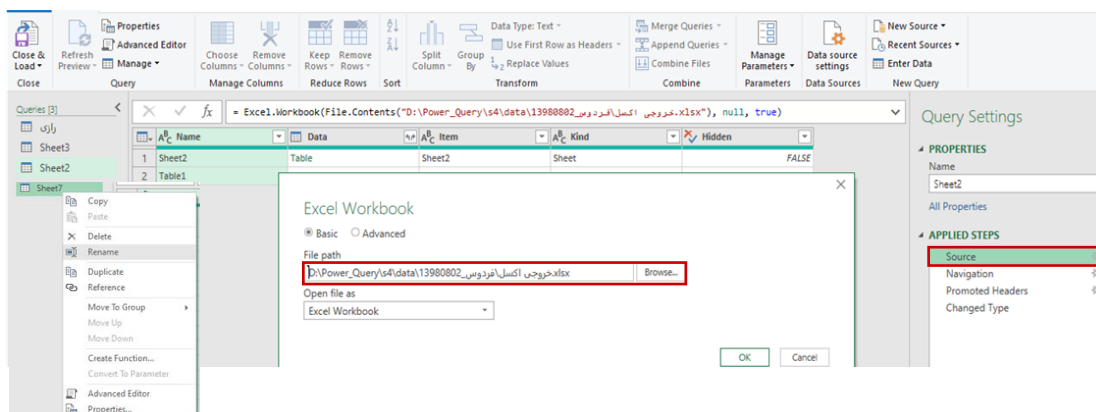


ستون 1	ستون 2	ستون 3	ستون 2
دیتا	دیتا		
دیتا	دیتا		
دیتا	دیتا		
دیتا		دیتا	دیتا
دیتا		دیتا	دیتا

در جلسه‌ی قبل درباره‌ی Lists اشاراتی کردیم که با Drill Down و Add As New Query می‌توانیم از سرستون‌ها یک List بسازیم و با یک Mapping در Append استفاده کنیم. حالا اگر در یک جدول نام یک ستون یک چیز باشد و در جدول دیگر نام ستون یک چیز دیگر، چه باید کرد؟

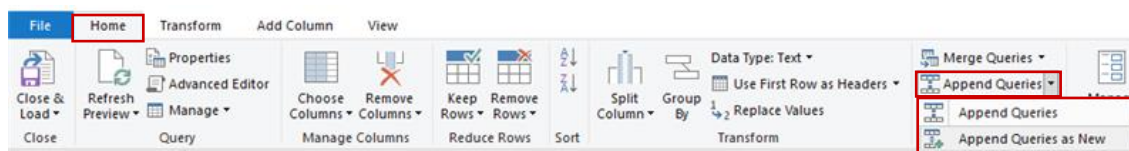
می‌توانیم به‌صورت دستی Rename کنیم، اما راه‌حل حرفه‌ای‌تر این است که سراغ زبان M برویم و از List ها کمک بگیریم. (در واقع خودکارسازی انجام می‌شود) هر Query ای که داریم، اگر از جنس Excel باشد و از آن آمده باشد، یک منبع داده (Data Source) دارد که هم می‌توانیم آن را ببینیم و هم تغییر دهیم. از بخش Applied Steps روی Source

کلیک می‌کنیم (یا روی چرخ‌دنده‌ی کنار آن). صفحه‌ای باز می‌شود که مسیر فایل (file path) را نشان می‌دهد و همچنین نوع فایل را مشخص می‌کند.



در آن مسیر، معمولاً نام فایل هم وجود دارد؛ پس می‌توانیم از اطلاعات نام فایل متوجه شویم که داده از کدام شرکت آمده و سپس نام Query را Rename کنیم. (وقتی تعداد Query ها زیاد باشد و نام‌گذاری مناسبی نداشته باشند، این کار کمک زیادی می‌کند).

در تب Home در بخش Combine گزینه Append Queries را داریم که دو بخش دارد:



- Append Queries as New : زمانی که می‌خواهیم ترکیب چند Query را در یک Query جدید بیاوریم.
- Append Queries : زمانی که در یک Query هستیم و می‌خواهیم بقیه‌ی Query ها و جداول را در همان اضافه کنیم.

کادر Append باز می‌شود. در آن گزینه‌ی Three or More Tables را می‌بینیم که باید انتخاب کنیم. هر جدولی که بخواهیم از سمت راست اضافه می‌کنیم. (باید توجه کنیم که هر بار که روی Add کلیک کنیم، آن جدول اضافه می‌شود. در واقع برنامه این اجازه را به ما می‌دهد، پس باید دقت کنیم که هر جدول را آیا یک‌بار می‌خواهیم اضافه کنیم یا بیشتر.)

Append

Concatenate rows from three or more tables into a single table.

☐ Two tables ☒ Three or more tables

Available tables

راز
سها
فردوس

Tables to append

راز
سها
فردوس

Add >>

OK

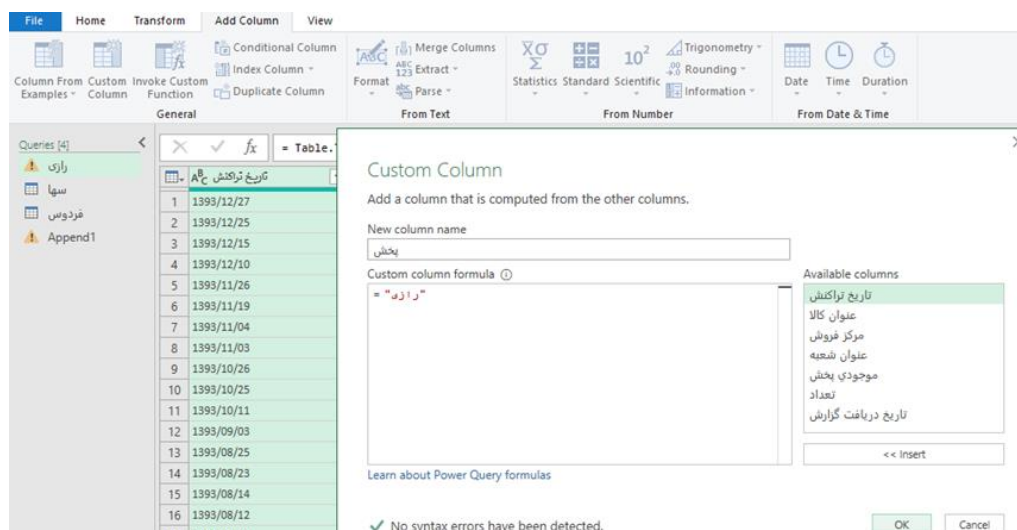
Cancel

اکنون باید مطمئن شویم که Append درست انجام شده است یا خیر و باید بررسی کنیم که جمع سطرهاهای هر چند جدول که آوردیم، با جمع سطرهاهای جدول Appended برابر باشد. اگر تعداد ستون‌ها در جداول اولیه مثلاً ۷ ستون بوده اما در نتیجه‌ی Append ۸ ستون نمایش داده شود، یعنی مشکلی وجود دارد.

در بالای محیط Power Query، از بخش Column Profiling می‌توانیم متوجه خطاهای کلی شویم؛ مثلاً ببینیم کدام ستون‌ها دارای خطا هستند یا خیر. سه راه برای پیدا کردن مشکل وجود دارد:

- می‌توان عنوان ستون‌های هر جدول را نگاه کرد و ستونی را که در نام با بقیه تفاوت دارد، Rename کرد. در این صورت جدول جدید الحاق شده هم ۷ ستون خواهد داشت و مشکل اختلاف ستون‌ها حل می‌شود. از پایین صفحه نیز می‌توان دید که تعداد ستون‌ها برابر و درست شده است.
- راه دوم استفاده از زبان M است. در این روش، با نوشتن کد، می‌توانیم هر ستونی که نامش مثلاً "نام کالا" بوده را به "عنوان کالا" تغییر دهیم و به نوعی یکسان‌سازی انجام دهیم.

- راه سوم از سمت خود Append قابل تشخیص است. باید نشانه‌ای از هر جدول، در جدول Appended وجود داشته باشد تا بدانیم داده از کدام جدول آمده است. اگر چنین ستونی نداریم، می‌توانیم آن را به صورت دستی اضافه کنیم. برای این کار، در هر جدول یک ستون جدید ایجاد می‌کنیم. از مسیر Add Column و قسمت Custom Column نام ستون جدید را می‌نویسیم و درون دابل کوتیشن، نام شرکت را قرار می‌دهیم.

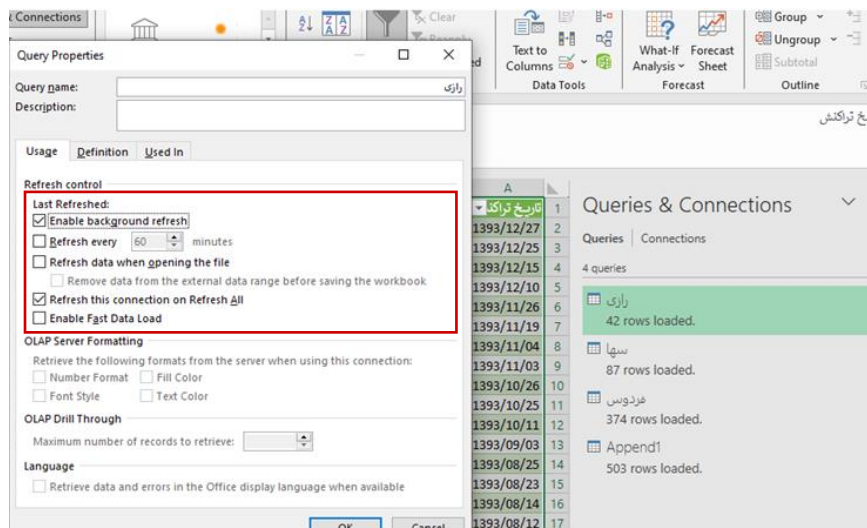


البته این روش ممکن است زمان‌بر باشد، مخصوصاً اگر تعداد جدول‌ها زیاد باشد. اما می‌توان اقدام دیگری نیز انجام داد: وقتی Folder را در Query ایمپورت می‌کنیم، به صورت خودکار یک ستون اضافه می‌شود که نام Sheet، نام Table یا File Name را می‌خواند. در نتیجه، این اطلاعات به عنوان یک ستون در جدول قرار می‌گیرند. (این موضوع را در جلسه‌ی پنجم به طور کامل توضیح می‌دهیم).

توجه به دو نکته‌ی حرفه‌ای اهمیت دارد:

۱. آیا می‌توان کاری کرد که هر فایل ستونی به نام خود Query داشته باشد؟ این کار ممکن است، اما پیچیدگی دارد و گاهی انجام‌ناپذیر است؛ زیرا باید Code های خاصی به آن اضافه کنیم.
۲. راه‌حل دیگر همان Import Folder است (شرح در جلسه پنجم) در این روش، نام Sheet یا File به صورت خودکار خوانده می‌شود و در قالب یک ستون در جدول قرار می‌گیرد. حال با روش سوم، یعنی اضافه کردن یک ستون جدید (مثلاً نام پخش یا شرکت)، می‌توانیم بررسی کنیم که کدام ستون‌های دیگر دارای داده‌ی null هستند. همچنین اگر ستون "عنوان کالا" را دوباره Rename کنیم و به "کالا" برگردانیم،

در اکسل‌هایی که لود می‌شوند، می‌توانیم در بخش Properties گزینه‌ی Refresh خودکار را غیرفعال کنیم. زیرا فایل Appended خودش به‌صورت خودکار Refresh می‌شود، بنابراین نیازی نیست شیت‌های دیگر را دوباره بارگذاری کنیم.



همچنین در زمان Load کردن می‌توان گزینه‌ی Load To را انتخاب کرد تا فقط Connection برقرار شود و تنها نتیجه‌ی Append نمایش داده شود.

البته در Power Query در Power BI می‌توان گزینه‌ای را انتخاب کرد که جلوی Refresh شدن کوئری را بگیرد و فقط یک Connection برقرار کند (که در پاور کوئری اکسل نیز این امکان بود از ابتدا که می‌خواستیم Load کنیم، Load to ... را بزنیم و گزینه Only Connection را تیک بزنیم)

یک نکته‌ی دیگر اینکه اگر به نوار Formula Bar دقت کنیم، نام تابع ما Table.Combine است و در آن لیستی از جداولی که به هم Append شده‌اند، قابل مشاهده است.



اگر داده و جدول جدیدی بدست ما برسد و بخواهیم به این جدول اضافه کنیم، باید در Source کوئری موردنظر وارد شویم و نام شیت یا فایل جدید را اضافه کنیم.

دستور Merge

Merge به معنای ادغام کردن است. هدف اصلی آن اضافه کردن ستون‌ها به جدول‌هاست، اما ممکن است در سطرها هم تغییراتی رخ دهد.

کسانی که با Power Pivot در Excel و یا با Data Model در Power BI کار می‌کنند، و به‌طور کلی هرگونه مدل‌سازی داده انجام می‌دهند، می‌دانند که مفهومی به نام Relationship وجود دارد و به ما کمک می‌کند بین جدول‌ها، براساس فیلدهای مشترک، ارتباطی تعریف کنیم تا بتوانیم داده‌های مرتبط را از جدول‌های دیگر فراخوانی کنیم.

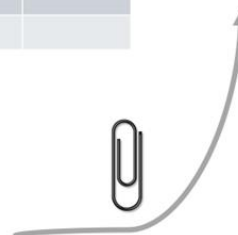
باید توجه داشت که Relationship ها در واقع پشت صحنه‌ی همان Merge عمل می‌کنند و در اصل جدول‌های ادغام‌شده یا Expanded Table هایی می‌سازند که ادغام افقی انجام می‌دهند.

اما نکته‌ی مهم این است که اگر با Relationship ها آشنا هستیم، به این معنا نیست که دیگر نیازی به Merge نداریم. در واقع، برخی کارها با Relationship ها انجام می‌شوند و برخی دیگر فقط با Merge قابل انجام هستند.

Merge ابزاری برای اضافه کردن ستون‌ها است، اما می‌تواند به دلیل تکراری بودن داده‌ها در سطرها، جدول دوم، باعث افزایش تعداد سطرها در جدول نهایی نیز بشود. برای مثال، فرض کنید دو جدول آبی و سبز رنگ داریم که در آن‌ها داده‌های تکراری وجود دارد.

توضیحات	نوع تسویه	عنوان شعبه	تعداد سفارش	کد مشتری	کد کالا	تاریخ
--	نقدی	مرکزی	1	523555	12001	1403/01/28
	نقدی	مرکزی	1	547900	12001	1403/01/28
	نقدی	مرکزی	1	301458	19012	1403/01/28
	نقدی	مرکزی	1	125120	14005	1403/01/28

مشخصات	قیمت واحد	نام کالا	کد کالا
50 تایی	25000000	قرص ظرفشویی finish	12001
1 لیتری	3910000	مایع ظرفشویی پرل	12002
نیم لیتری	2950000	جلا دهنده پرل	12003
	2870000	نمک پرل	12004
نیم لیتری	3500000	جلا دهنده finish	12005



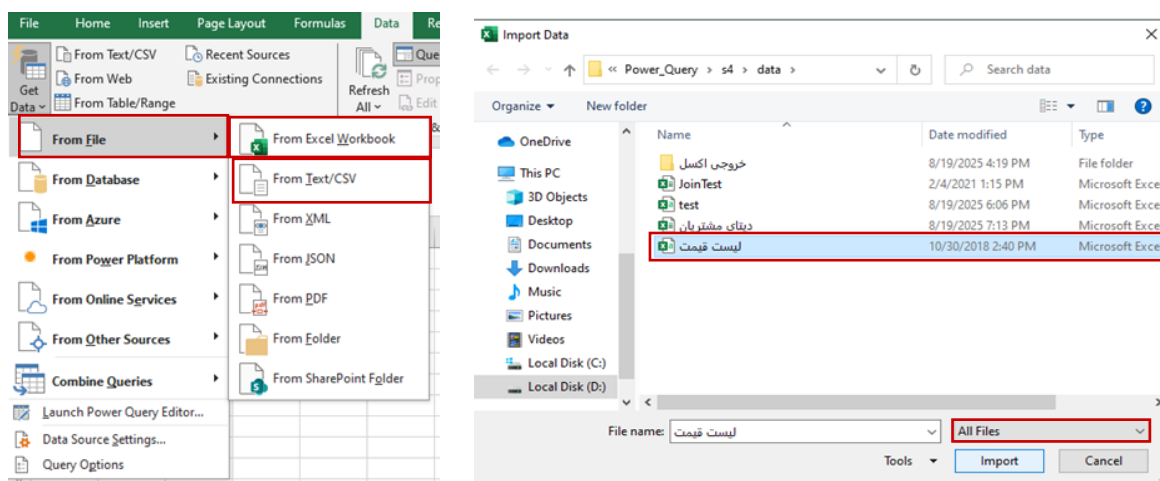
همان جدول Append قبلی را فرض کنید، اما این بار ستون «قیمت واحد» را نداریم. پس باید داده‌ی مربوط به آن را وارد کنیم. (در این مثال، این داده‌ها از فایل CSV آمده‌اند). وقتی فایل CSV را دریافت می‌کنیم (مثلاً خروجی از یک نرم‌افزار دیگر باشد)، ممکن است به‌صورت جداگانه یا در کنار سایر فایل‌های Excel ذخیره شده باشد. در بسیاری از سیستم‌ها، خروجی‌ها در قالب CSV ایجاد می‌شوند. ما زمانی از این نوع فایل استفاده می‌کنیم که به هر دلیلی داده‌ها در این فایل ذخیره

شده و به همین صورت هم بدست ما رسیده است و یا به دلیل آنکه فایل اکسل ظرفیت خروجی آن داده‌ها را نداشته است. (تا این لحظه تعداد سطرهای اکسل ۱۰۴۸.۵۷۶ سطر است) در CSV داده‌ها سبک‌تر و سریع‌تر هستند.

البته در Power Pivot اکسل می‌توان داده‌های نامحدود وارد کرد، و در Power BI نیز به اندازه‌ی ظرفیت حافظه، داده‌ها قابلیت Load شدن دارند. اما خود Excel به‌صورت مستقیم چنین ظرفیت نامحدودی ندارد.

برای وارد کردن فایل CSV دو راه وجود دارد:

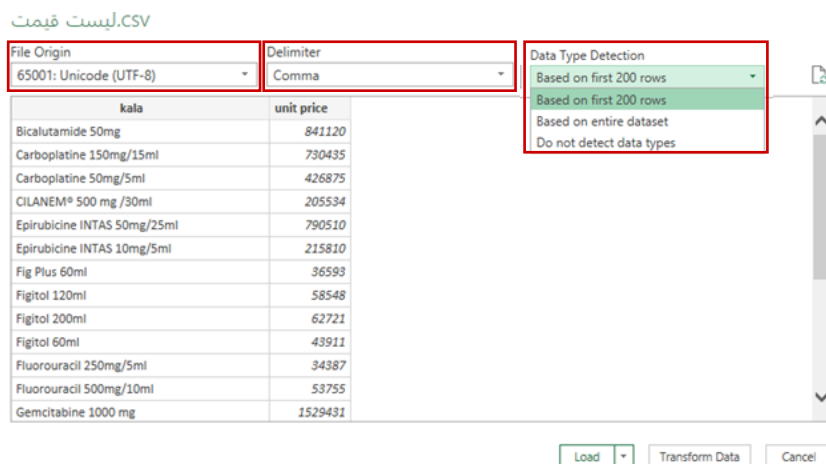
- از بخش Get Data > From Text/CSV
- یا از Get Data > From Excel Workbook و سپس تغییر File Type به All Files تا همه‌ی فایل‌ها از جمله CSV نمایش داده شوند.



در پنجره‌ای که ظاهر می‌شود، سه گزینه برای تنظیم داریم:

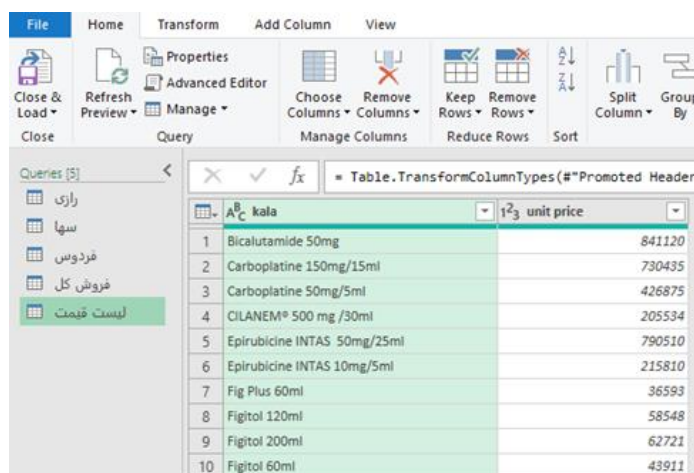
- File Origin : نوع رمزگذاری فایل که معمولاً به‌صورت خودکار تشخیص داده می‌شود.
- Delimiter : جداکننده‌ی داده‌ها (مثلاً Comma یا همان ویرگول).
- Data Type Detection : نحوه‌ی تشخیص نوع داده (می‌تواند فقط بر اساس ۲۰۰ سطر اول، کل داده‌ها، یا بدون

تشخیص خودکار انجام شود)



در فایل‌های CSV معمولاً داده‌ها با کاما از هم جدا می‌شوند. بنابراین Power Query خودش جداکننده را تشخیص

می‌دهد و داده‌ها را به صورت ستونی در جدول نمایش می‌دهد.



	kala	unit price
1	Bicalutamide 50mg	841120
2	Carboplatine 150mg/15ml	730435
3	Carboplatine 50mg/5ml	426875
4	CILANEM® 500 mg /30ml	205534
5	Epirubicine INTAS 50mg/25ml	790510
6	Epirubicine INTAS 10mg/5ml	215810
7	Fig Plus 60ml	36593
8	Figitol 120ml	58548
9	Figitol 200ml	62721
10	Figitol 60ml	43911

انجام عملیات Merge

وقتی جدول «لیست قیمت» را در Power Query وارد کردیم، ممکن است یکی یا چند ستون به عنوان داده‌های

ناقص یا مفقود شناسایی شوند. در اینجا نام ستون‌ها مهم نیست، بلکه داده‌های موجود در آن‌ها اهمیت دارند.

در بخش Combine دو گزینه برای ادغام داریم:

۱. Merge Queries

۲. Merge Queries as New

FileHomeTransformAdd ColumnView

Close & LoadClose

Refresh Preview

PropertiesAdvanced EditorManageQuery

Choose ColumnsManage Columns

Remove RowsRemove Columns

Keep RowsReduce Rows

Sort

Split ColumnByTransform

Data Type: TextUse First Row as HeadersReplace Values

Merge QueriesMerge Queries as NewCombineParameters

New SourceData source settingsData Sources

Recent SourcesEnter DataNew Query

Queries5<

×

✓

fx

= Table.Combine(فرودسها و فرودسها)

▼

تعداد	موجودی بخش	عنوان شعبه	مرکز فروش	عنوان کالا	تاریخ تراکنش
246	تهران	داروسازی نوین	Fig Plus 60ml	1399/12/27	1
6	اصفهان	داروسازی نوین	Figitol 120ml	1399/12/25	2
0	شیراز	داروسازی نوین	Gynophilus	1399/12/15	3
29	اهواز	داروسازی نوین	Gynophilus	1399/12/10	4

Query Settings

PROPERTIES

Name

فروشن کل

All Properties

معمولاً مورد مهم‌تر Merge Queries است، (بر خلاف Append، نیازی به ایجاد Query جدید نیست). چون

می‌خواهیم در همان جدولی که هستیم، ستون‌های جدید را اضافه کنیم.

در اینجا قصد داریم ستون‌هایی که نداریم را از یک جدول دیگر، به همان جدول فعلی اضافه کنیم. وقتی پنجره‌ی

Merge باز می‌شود، باید دو ستون مشترک بین جدول‌ها داشته باشیم تا بتوانند بر اساس آن ادغام شوند.

اگر نام ستون‌ها یکسان نباشد، مشکلی نیست، فقط باید مشخص کنیم که کدام ستون در جدول اول با کدام ستون در جدول

دوم مرتبط است.

Merge

Select a table and matching columns to create a merged table.

فروش کل

تاریخ ترک‌شدن	عنوان کالا	مرکز فروش	عنوان شعبه	موجودی پخش	تعداد	تاریخ دریافت گزارش	پخش
1393/12/27	Fig Plus 60ml	داروسازی نوین	تهران		246	1393/12/30	رای
1393/12/25	Figitol 120ml	داروسازی نوین	اصفهان		67	1393/12/30	رای
1393/12/15	Gynophillus	داروسازی نوین	شیراز		0	1393/12/30	رای
1393/12/10	Gynophillus	داروسازی نوین	اهواز		29	1393/12/30	رای
1393/11/26	Gynophillus	داروسازی نوین	مشهد		0	1393/11/30	رای

لیست قیمت

kala	unit price
Bicalutamide 50mg	841120
Carboplatine 150mg/15ml	790435
Carboplatine 50mg/5ml	426875
CILANEM® 500 mg /30ml	205534
Epirubicine INTAS 50mg/25ml	790510

Join Kind
Left Outer (all from first, matching from second)

☐ Use fuzzy matching to perform the merge

Fuzzy matching options

☒ The selection matches 503 of 503 rows from the first table.

OK Cancel

در پایین صفحه، نتیجه نمایش داده می‌شود که نشان می‌دهد از مجموع سطرهای جدول، چه تعدادی Match شده‌اند. پس از انجام عملیات Merge، در جدول یک ستون جدید اضافه می‌شود که قابلیت Expand دارد. با کلیک روی آیکن Expand (علامت دو فلش ↔)، می‌توانیم انتخاب کنیم که از جدول جدید، چه ستون‌هایی را وارد جدول اصلی کنیم. همچنین از تب Transform هم می‌توان این Expand را انجام داد.

در پایین کادر Expand گزینه‌ای وجود دارد که می‌توانیم تعیین کنیم آیا نام جدول منبع به عنوان پیشوند (Prefix) به ستون‌های جدید اضافه شود یا خیر. برای مثال، اگر در جدول اصلی ستونی به نام "قیمت" داریم و در جدول جدید هم ستونی با همین نام وجود دارد، می‌توانیم در قسمت Prefix بنویسیم مثلاً "m_ قیمت" تا بدانیم این ستون از عملیات Merge آمده است.

Expand قیمت

Select the columns to expand.

Search Columns

☒ (Select All Columns)

☐ kala

☒ unit price

Default column name prefix (optional)

m_

OK Cancel

Query Settings

Table

Properties

Name

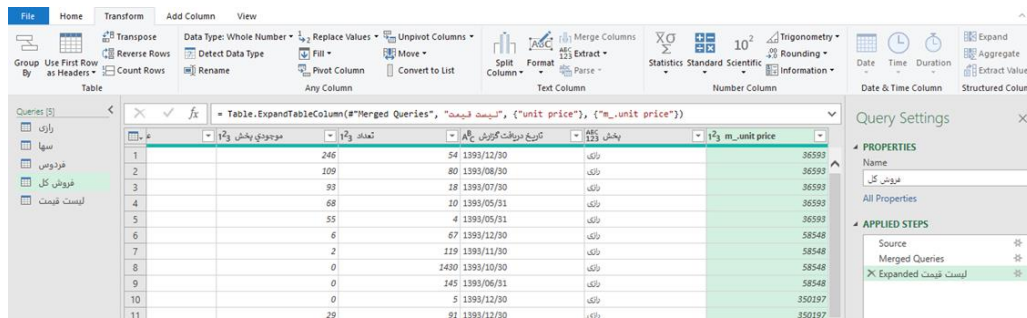
فروش کل

APPLIED STEPS

Source

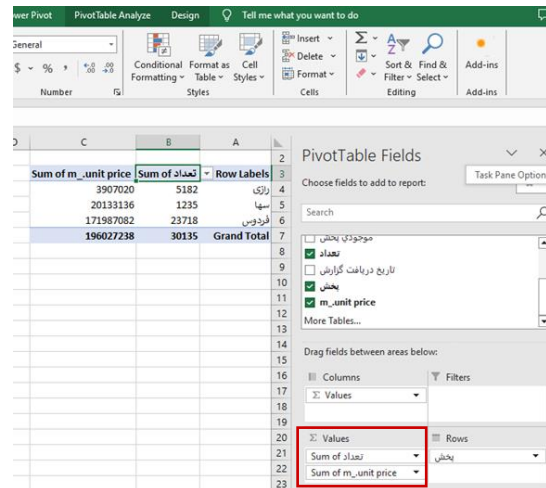
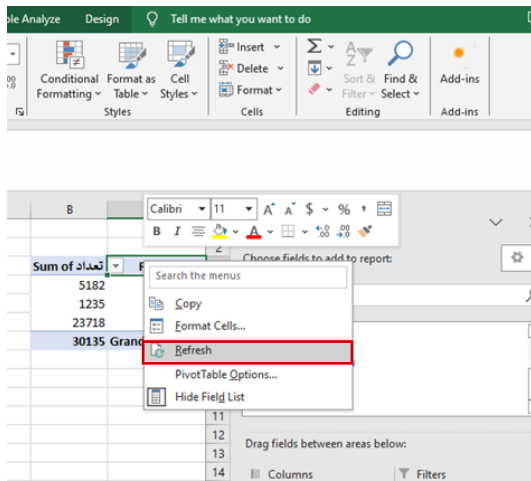
Merged Queries

پس از انتخاب ستون‌ها و تأیید، ستون‌های موردنظر (مثلاً "قیمت") به جدول اصلی اضافه می‌شوند.



سپس اگر داده را Load کنیم، در Excel خروجی، جدول Merged نمایش داده می شود.

در Pivot Table هم اگر Refresh بزنیم، ستون جدید (مثلاً قیمت) اضافه می شود و می توان گزارش هایی بر اساس آن تنظیم کرد، مثلاً محاسبه ی مجموع مبلغ فروش.



سؤال مهم این است که منشاء Merge چیست و از کجا آمده است؟

پیش از به وجود آمدن Power Query، در دنیای پایگاه های داده (Databases) مفهومی به نام Join در زبان SQL وجود داشت. در واقع، Merge در Power Query همان نقش Join را در SQL Server ایفا می کند؛ یعنی ترکیب داده ها بر اساس فیلدهای مشترک بین جدول ها.

SQL Join Types

1. Inner Join
2. Left Outer Join
3. Right Outer Join
4. Full Outer Join
5. Cross Join

(Cross Join در Power Query وجود ندارد و اگر بخواهیم چنین ادغامی انجام دهیم، باید آن را به صورت دستی ایجاد کنیم)

اما در Power Query، انواع Join Types کمی متفاوت هستند.

Power Query Join Types

1. Left Outer
2. Right Outer
3. Full Outer
4. Inner
5. Left Anti
6. Right Anti

انواع Merge (Join Types) در Power Query

• Inner Join

فقط رکوردهایی را نمایش می دهد که مقدار مشترک در هر دو جدول دارند. یعنی فقط داده هایی که هم در جدول A و هم در جدول B وجود دارند، در نتیجه ی Merge نمایش داده می شوند.

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Heisenberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
Williams	NULL		



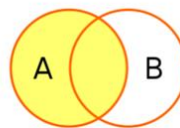
↓

Employee.LastName	Employee.DepartmentID	Department.DepartmentName
Robinson	34	Clerical
Jones	33	Engineering
Smith	34	Clerical
Heisenberg	33	Engineering
Rafferty	31	Sales

• Left Outer Join

همه ی رکوردهای جدول A (جدول اول) و فقط رکوردهای مشترک آن با جدول B را می آورد. داده هایی که در جدول A هستند اما در B نیستند، همچنان در خروجی دیده می شوند، ولی ستون های مربوط به B مقدار Null خواهند داشت.

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Heisenberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
Williams	NULL		



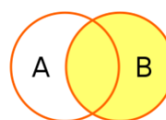
Employee.LastName	Employee.DepartmentID	Department.DepartmentName
Jones	33	Engineering
Rafferty	31	Sales
Robinson	34	Clerical
Smith	34	Clerical
Williams	NULL	NULL
Heisenberg	33	Engineering

Right Outer Join •

مشابه Left Outer است، اما جهت عکس دارد. همه‌ی رکوردهای جدول B و اشتراک آن با A را برمی‌گرداند.

در واقع، اگر جای جدول‌ها را در Left Outer عوض کنیم (یعنی اول B و بعد A را انتخاب کنیم)، همان Right Outer به‌دست می‌آید.

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Heisenberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
Williams	NULL		



Employee.LastName	Employee.DepartmentID	Department.DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Heisenberg	33	Engineering
Rafferty	31	Sales
NULL	NULL	Marketing

Full Outer Join •

همه‌ی رکوردهای هر دو جدول A و B را نمایش می‌دهد؛ چه مشترک باشند و چه نباشند. در نتیجه، اگر مقداری در

یک جدول وجود داشته باشد ولی در دیگری نه، باز هم در خروجی نهایی نمایش داده می‌شود.

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Heisenberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
Williams	NULL		



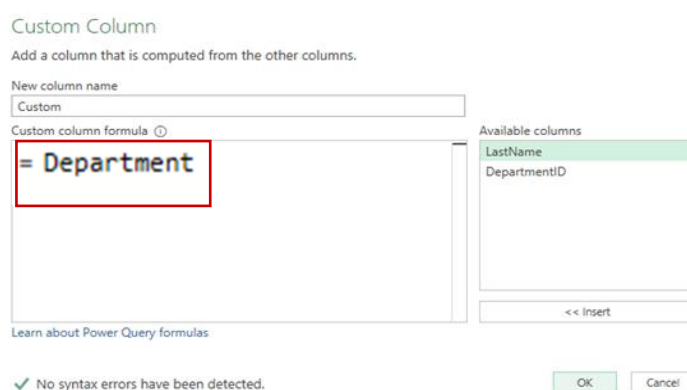
Employee.LastName	Employee.DepartmentID	Department.DepartmentName
Smith	34	Clerical
Jones	33	Engineering
Robinson	34	Clerical
Williams	NULL	NULL
Heisenberg	33	Engineering
Rafferty	31	Sales
NULL	NULL	Marketing

• ایجاد Cross Join در Power Query

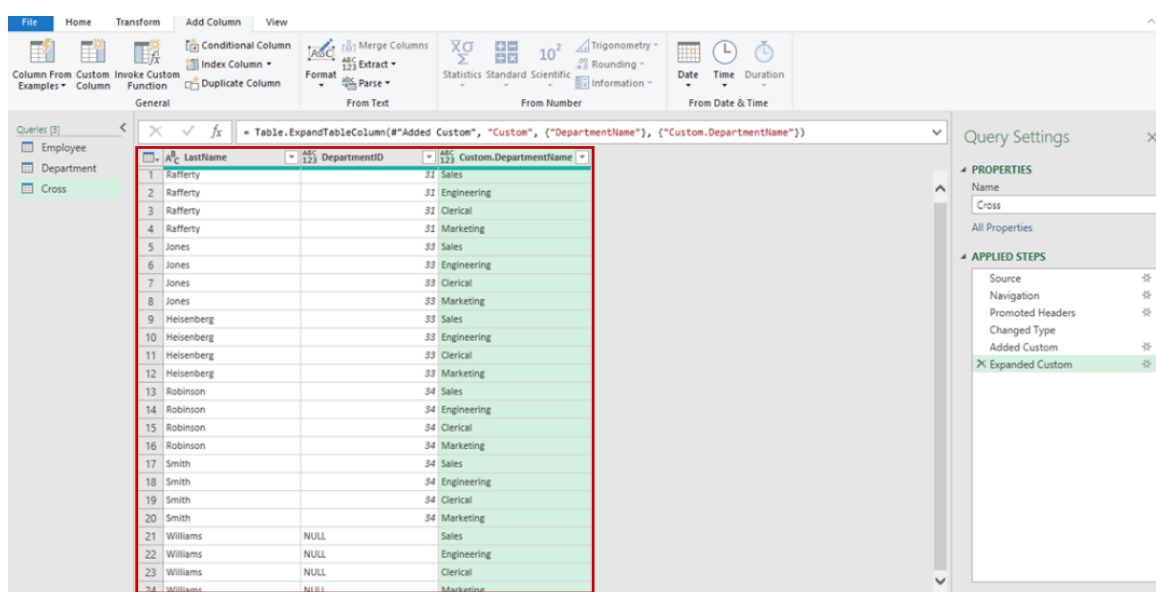
در برخی مواقع، لازم است تمام ترکیب‌های ممکن بین دو جدول را بسازیم، یعنی هر سطر از جدول اول با تمام سطرهاى جدول دوم ترکیب شود. در SQL به این نوع ادغام، Cross Join گفته می‌شود. در Power Query دستور مستقیم برای Cross Join وجود ندارد، اما می‌توان آن را به صورت دستی ایجاد کرد.

ابتدا از جدول اول مثلاً جدول (Department) یک نسخه‌ی تکراری (Duplicate) می‌گیریم. سپس از نوار ابزار، مسیر Add Column → Custom Column را انتخاب می‌کنیم. در بخش فرمول (Formula Bar) می‌نویسیم:

با این کار، در واقع جدول Department را به عنوان یک ستون جدید به جدول فعلی اضافه می‌کنیم.



نتیجه این است که یک ستون جدید حاوی داده‌های جدول Department به دست می‌آید که آماده‌ی Expand شدن است. حالا وقتی این ستون را Expand کنیم، تعداد سطرهاى جدول نهایی برابر خواهد شد با حاصل ضرب تعداد سطرهاى هر دو جدول: مثلاً اگر جدول اول ۴ سطر و جدول دوم ۶ سطر داشته باشد، نتیجه‌ی Cross Join شامل ۲۴ سطر خواهد بود.

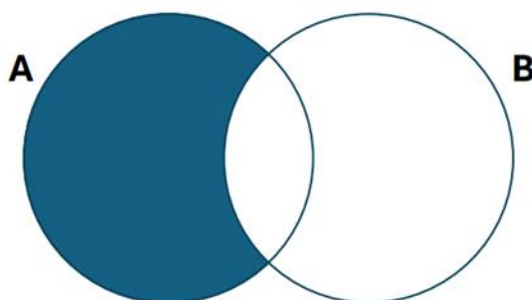


اما روش Cross Join چه فایده‌ای دارد و کاربرد آن چیست؟

فرض کنید می‌خواهیم برای پروازهای مختلف یک Airline، خلبان‌ها را به‌صورت ترکیبی تخصیص دهیم. با Cross Join می‌توان تمام حالت‌های ممکن را ساخت، یعنی تمام خلبان‌ها در مقابل تمام پروازها لیست بشود. گاهی هم از Cross Join برای تحلیل داده‌های ناقص استفاده می‌کنیم. مثلاً اگر بخواهیم بدانیم کدام کالاها در کدام ماه‌ها فروش نرفته‌اند، می‌توانیم تمام ماه‌ها و کالاها را با Cross Join ترکیب کنیم. سپس یک ستون محاسباتی (Calculated Column) اضافه کنیم تا مشخص کند در هر ترکیب، عدد فروش وجود دارد یا صفر است. در نهایت، مواردی که مقدار فروششان صفر است را روی نمودار نمایش دهیم تا کالاهای فروش‌نرفته مشخص شوند. در Power Query دو نوع Join دیگر وجود دارد که مخصوص پیدا کردن داده‌های غیرمشترک هستند:

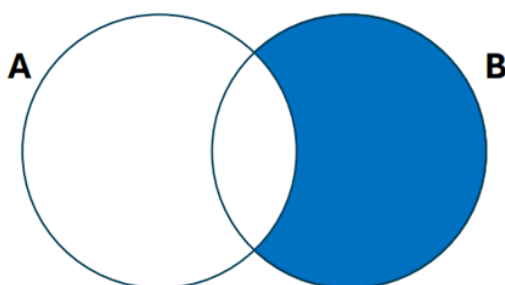
• Left Anti Join

فقط رکوردهایی را نمایش می‌دهد که در جدول A وجود دارند اما در جدول B نیستند. کاربرد آن در پیدا کردن داده‌های کمبود یا آیتم‌هایی هستند که تطابق ندارند. برای مثال اگر فهرستی از کارکنان داریم و می‌خواهیم افرادی را پیدا کنیم که هنوز واحد سازمانی (Department) ندارند، از Left Anti استفاده می‌کنیم.



• Right Anti Join

برعکس مورد قبلی است. فقط داده‌هایی را نشان می‌دهد که در جدول B وجود دارند ولی در جدول A نیستند. برای مثال اگر فهرستی از کالاها داریم که در جدول دوم موجودی انبار است، با Right Anti می‌توانیم کالاهایی را پیدا کنیم که در انبار هستند ولی در لیست فروش وجود ندارند.



مثال تحلیلی: تشخیص مشتریان غیرفعال

فرض کنید لیستی از مشتریان داریم که کالای A را خریده‌اند و لیست دیگری از مشتریانی که کالای B را خریده‌اند. با استفاده از Left Anti Join می‌توانیم بفهمیم کدام مشتریان کالای A را خریده‌اند ولی کالای B را نه. این اطلاعات می‌تواند برای پیشنهاد تخفیف یا بازاریابی هدفمند بسیار مفید باشد.

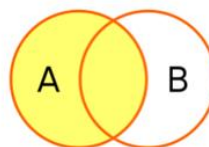
نکته در مورد کوئری‌ها: وقتی تعداد Query ها زیاد می‌شود، برای نظم‌دهی بهتر می‌توانیم روی هر Query کلیک راست کنیم و گزینه‌ی New Group را انتخاب کنیم. سپس Query های مرتبط را در آن گروه قرار دهیم تا ساختار پروژه تمیزتر و قابل مدیریت‌تر شود.

• Fuzzy Matching و Right Join

در واقع، Right Join از مفاهیم پایه‌ای در SQL است و در Power Query نیز عملکردی مشابه دارد. در Right Join تمام رکوردهای جدول دوم (Table B) نمایش داده می‌شوند و فقط داده‌های مرتبط از جدول اول (Table A) به آن‌ها متصل می‌شوند. اگر در جدول دوم رکوردهایی وجود داشته باشند که در جدول اول معادلی ندارند، مقدار ستون‌های مربوط به جدول اول برای آن رکوردها Null خواهد بود. در Power Query نیز همین رفتار وجود دارد، و می‌توان با انتخاب نوع Join Type = Right Outer به همین نتیجه رسید.

توجه به کد یا ساختار SQL Join به درک بهتر عملکرد Merge در Power Query کمک می‌کند، چون منطق یکی است و تنها رابط کاربری متفاوت است.

Employee table		Department table	
LastName	DepartmentID	DepartmentID	DepartmentName
Rafferty	31	31	Sales
Jones	33	33	Engineering
Heisenberg	33	34	Clerical
Robinson	34	35	Marketing
Smith	34		
Williams	NULL		



```
SELECT * FROM Employee
LEFT JOIN Department
ON Employee.DepartmentID = Department.DepartmentID
```


Fuzzy Matching

فرض کنید سه جدول داریم: جدول مشتریان، درخواست‌ها، تطبیق

در حالت عادی، هنگام Merge اگر نام شرکت‌ها دقیقاً یکسان نباشند، ادغام انجام نمی‌شود و در ستون‌های مربوط

مقدار Null ظاهر می‌شود.

Merge

Select tables and matching columns to create a merged table.

درخواست‌ها

نام مشتری	تاریخ	تعداد درخواست
نو آوران فن آوازه	1399/5/02	3
IranCell	99/05/02	4
IranCell	1399/05/03	2
رایتل	1399/05/2	1
Knauf	1399/05/03	3

مشتریان

کد مشتری	نام مشتری	نوع مشتری	گروه مشتری
23	دیجی‌کان	حقوق	فروش اینترنتی
24	ایرانسل	حقوق	اپراتور تلفن همراه
25	رایتل	حقوق	اپراتور تلفن همراه
26	کتاب ایران	حقوق	ساخته‌های
27	ایران رهجو	حقوق	گزارشی کامپیوتر

Join Kind

Left Outer (all from first, matching from second)

☐ Use fuzzy matching to perform the merge

Fuzzy matching options

☒ The selection matches 8 of 14 rows from the first table.

File Home Transform Add Column View

Close & Load Refresh Preview Advanced Editor Manage Query

Choose Columns Remove Columns Keep Rows Remove Rows Sort Split Column Group By Data Type: Text Use First Row as Headers Replace Values Transform Merge Append Combine

Queries [3]

درخواست‌ها

مشتریان

جدول تطابق

Table.ExpandTableColumn("#Merged Queries", "مشتریان", "گروه مشتری", "گروه مشتری")

نام مشتری	تاریخ	تعداد درخواست	گروه مشتری
نو آوران فن آوازه	1399/5/02	3	null
IranCell	99/05/02	4	null
IranCell	1399/05/03	2	اپراتور تلفن همراه
رایتل	1399/05/2	1	null
Knauf	1399/05/03	3	گزارشی کامپیوتر
ایران رهجو	1399/5/03	2	null
ایران خوندو	99/05/04	5	null
علی زبانی	1399/05/04	2	وارد کننده
حمیده انتظاری	99/05/04	4	null
حمیده انتظاری	1399/05/05	4	آموزش
رایتل	1399/05/07	4	اپراتور تلفن همراه
رایتل	1399/5/08	2	اپراتور تلفن همراه
رایتل	1399/05/09	2	اپراتور تلفن همراه
رایتل	99/05/10	2	اپراتور تلفن همراه

گاهی تفاوت‌ها جزئی هستند، مثلاً وجود space اضافه، تفاوت در املای حروف فارسی و انگلیسی مثل "ایران خودرو"

و "Iran Khodro"، یا وجود کاراکترهای اضافی یا فاصله‌های ناخواسته، در چنین حالتی، Fuzzy Matching در Power

Query کمک می‌کند تا داده‌ها بر اساس شباهت تقریبی تطبیق داده شوند

با فعال شدن گزینهی Use fuzzy matching to perform the merge، چند گزینه‌ی مهم دیگر باز می‌شود

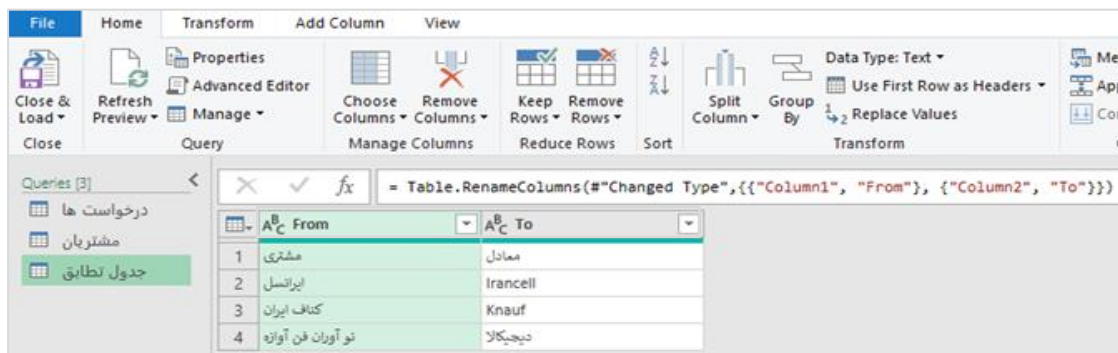
که رفتار تطبیق را کنترل می‌کنند:

- Ignore case : حروف بزرگ و کوچک را در مقایسه نادیده می‌گیرد.
- Match by combining text parts : فاصله‌ها و جداکننده‌ها را حذف می‌کند و سپس کلمات را به هم می‌چسباند تا مقایسه انجام شود. به عنوان مثال: "ایران خودرو" و "ایران خودرو"، که همه به عنوان یک مقدار شناخته می‌شوند.

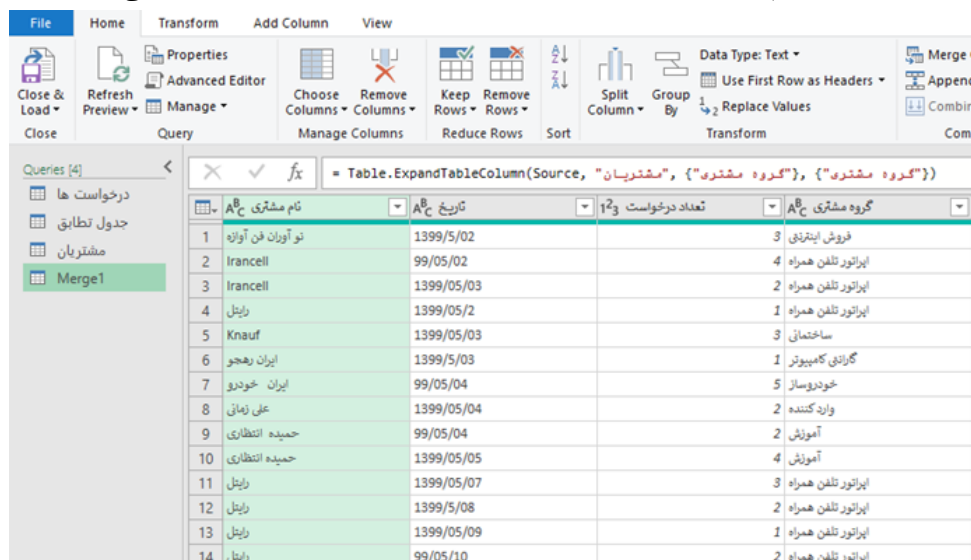
- **Similarity Threshold**: یعنی آستانه‌ی شباهت بین دو مقدار. به صورت پیش فرض روی ۰.۸ یا ۸۰٪ تنظیم شده است. یعنی اگر دو عبارت تا ۸۰٪ شبیه هم باشند، Power Query آن‌ها را برابر در نظر می‌گیرد. می‌توان این عدد را بسته به حساسیت کار تغییر داد.

- **Maximum number of matches**: تعیین می‌کند اگر چند مقدار مشابه در جدول دوم وجود داشت، چند مورد از آن‌ها را در نتیجه بیاورد. مثلاً اگر مقدارش را ۱ بگذاریم، فقط اولین تطبیق را برمی‌گرداند و اگر خالی بماند، همه‌ی تطبیق‌ها را می‌آورد (که ممکن است باعث تکرار داده‌ها شود).
- **Transformation Table**: این یکی از گزینه‌های بسیار کاربردی است. در این حالت، شما می‌توانید جدولی از قبل ساخته شده داشته باشید که در آن معادل‌های درست کلمات تعریف شده‌اند. این جدول را به عنوان Transformation Table به Merge معرفی می‌کنید. در نتیجه، Power Query هر بار که در داده‌ها مثلاً "Iracell" یا "ایرانسل" را ببیند، می‌داند که این دو یکی هستند.

توجه به این نکته ضروری است که اگر سرستون‌های جدول تطابق دقیقاً به نام‌های From و To نباشند، خطا (Error) رخ می‌دهد و Merge انجام نمی‌شود.



نهایتاً همه داده‌های در ستون نام مشتری را تشخیص داده و برا این اساس ستون جدید Merge می‌گردد.



نکته: اجرای Refresh در Query هایی که از Fuzzy Matching استفاده می‌کنند ممکن است باعث کندی سیستم شود، چون عملیات تطبیق فازی محاسبات زیادی انجام می‌دهد. توصیه می‌شود از Fuzzy Matching فقط در مواقع ضروری و با حجم داده‌های محدود استفاده شود.

در نتیجه، Fuzzy Matching ابزاری بسیار قدرتمند برای ادغام داده‌هایی است که شباهت ظاهری دارند ولی دقیقاً یکسان نیستند. این قابلیت مخصوصاً در پاک‌سازی داده‌ها و ادغام فایل‌های سازمانی یا گزارش‌های وارد شده از منابع مختلف بسیار مفید است.

جلسه پنجم

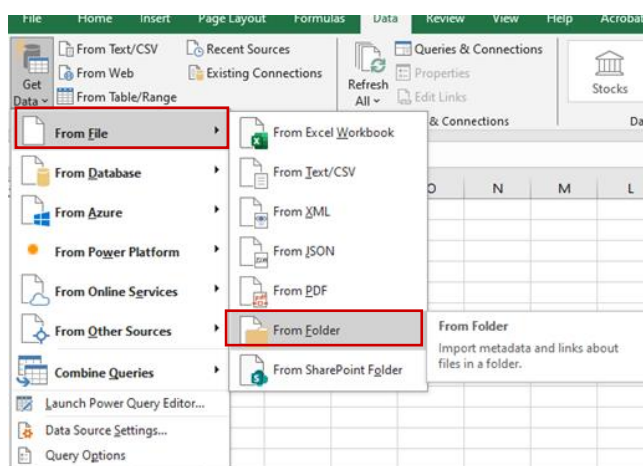
Import Folder

عناوینی که برای این بخش وجود دارد عبارتاند از:

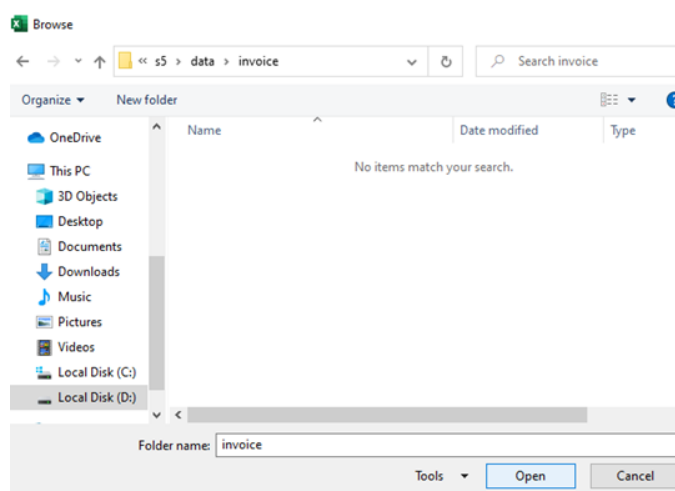
Batch Import/Bulk Import /Import Folder/ Combine

گاهی پیش می‌آید که نمی‌خواهیم فقط یک فایل را Import کنیم، بلکه نیاز داریم چند تا فایل Excel را با هم و به صورت یک جا Import کنیم.

در تب Data، گزینه‌ی Get Data > From Folder را می‌زنیم. پوشه را Import می‌کنیم؛ در این پوشه تمام فایل‌ها قرار دارند. ممکن است در آن هم فایل‌های xlsx باشند، هم csv، هم PowerPoint و فایل‌های متفاوت دیگر.



به مسیری می‌رویم که فولدر در آنجا قرار گرفته، وقتی به فولدر موردنظر برسیم و روی آن کلیک کنیم در صفحه Import دیگر فایلی را نشان نمی‌دهد، چون در انتخاب اولیه ما فولدر را Import کردیم. و این صفحه را نمایش می‌دهد.



در این صفحه ویژگی‌های فایل‌ها و مسیر آن‌ها نمایش داده می‌شود. حالا لیستی از فایل‌ها نمایش داده می‌شود که انواع مختلف دارند.

D:\Power_Query\s5\data\invoice

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes
Binary	newslier.jpg	.jpg	10/15/2025 11:37:09 PM	6/21/2020 8:01:02 PM	10/14/2025 11:01:53 PM	Record
Binary	new_pooryab_logo.png	.png	10/15/2025 10:34:04 PM	6/7/2018 1:23:06 PM	10/14/2025 11:01:53 PM	Record
Binary	PBI_Sheets.pptx	.pptx	10/15/2025 10:34:04 PM	4/28/2020 7:19:04 PM	10/14/2025 11:01:53 PM	Record
Binary	13980801_رازى.xlsx	.xlsx	10/16/2025 11:43:36 AM	1/26/2022 2:58:44 PM	10/14/2025 11:01:53 PM	Record
Binary	13980804_سپا.xlsx	.xlsx	10/16/2025 11:43:36 AM	8/25/2022 1:52:02 PM	10/14/2025 11:01:53 PM	Record
Binary	14001117_پخش دایا.xlsx	.xlsx	10/16/2025 11:43:36 AM	6/24/2021 6:17:28 PM	10/14/2025 11:01:53 PM	Record

Combine Load Transform Data Cancel

گزینه‌ی Load وجود دارد که اگر آن را انتخاب کنیم، همین جدول نمایش داده شده را به ما می‌دهد که شامل نام فایل، پسوند، تاریخ، ویژگی‌های فایل در Windows و مسیر فایل است. پس این گزینه به درد ما نمی‌خورد.

Combine کردن هم در شرایطی که فایل‌های غیر Excel در فولدر داریم پیشنهاد نمی‌شود، چون نمی‌دانیم که این فایل‌های Excel همه شیت‌هایشان هم‌شکل باشند و مشکلی نداشته باشند. پس این گزینه را هم نمی‌زنیم. بنابراین با انتخاب گزینه‌ی Transform، جدول را در محیط Power Query می‌آوریم.

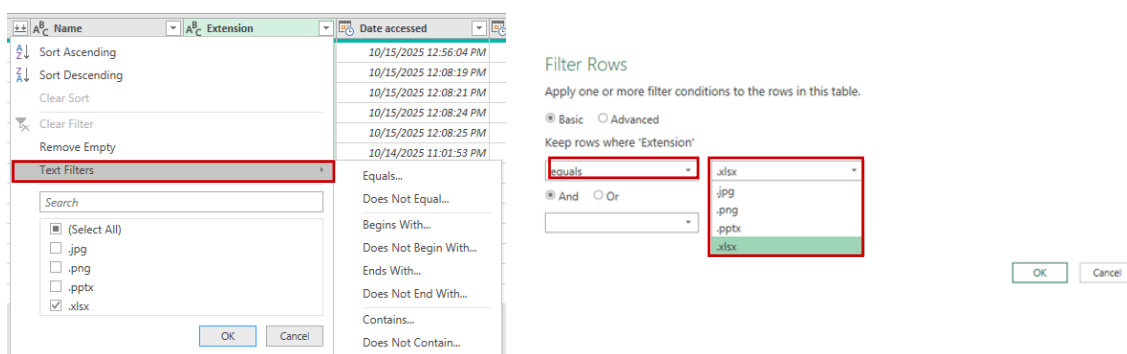
یک‌سری ستون داریم که در واقع ستون Content حالت Binary دارد و مثل لینک است (یعنی قابل کلیک)، و ستون Attribute که حالت Record دارد و هرکدام ویژگی‌هایی را نمایش می‌دهند. در پایین صفحه می‌توانیم آن ویژگی‌ها را ببینیم. Record ها نوع دیگری از داده هستند که در رده‌ی دوم اهمیت قرار دارند، (مثل List ها که در جلسه‌ی قبل اشاره شد و مهم بودند).

The screenshot shows the Power Query Editor interface. The main table displays file metadata with columns: Date accessed, Date modified, Date created, Attributes, and Folder Path. The 'Attributes' column is highlighted with a red box. Below the table, a detailed view of the selected file's attributes is shown, also with a red box around it.

Index	Date accessed	Date modified	Date created	Attributes	Folder Path
1	10/15/2025 11:48:56 AM	8/25/2022 1:48:52 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
2	10/15/2025 12:08:19 PM	8/26/2022 9:38:38 AM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
3	10/15/2025 12:08:21 PM	12/19/2018 11:01:44 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
4	10/15/2025 12:08:24 PM	8/25/2022 1:46:16 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
5	10/15/2025 12:08:25 PM	12/19/2018 11:01:58 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
6	10/14/2025 11:01:53 PM	8/25/2022 1:47:16 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
7	10/14/2025 11:01:53 PM	8/25/2022 1:48:08 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\
8	10/14/2025 11:01:53 PM	6/21/2020 8:01:02 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\invoice\
9	10/15/2025 11:48:58 AM	6/7/2018 1:23:06 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\invoice\
10	10/14/2025 11:01:53 PM	4/28/2020 7:19:04 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\invoice\
11	10/14/2025 11:01:53 PM	1/26/2022 2:58:44 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\invoice\
12	10/14/2025 11:01:53 PM	8/25/2022 1:52:02 PM	10/14/2025 11:01:53 PM	Record	D:\Power_Query\s5\data\invoice\

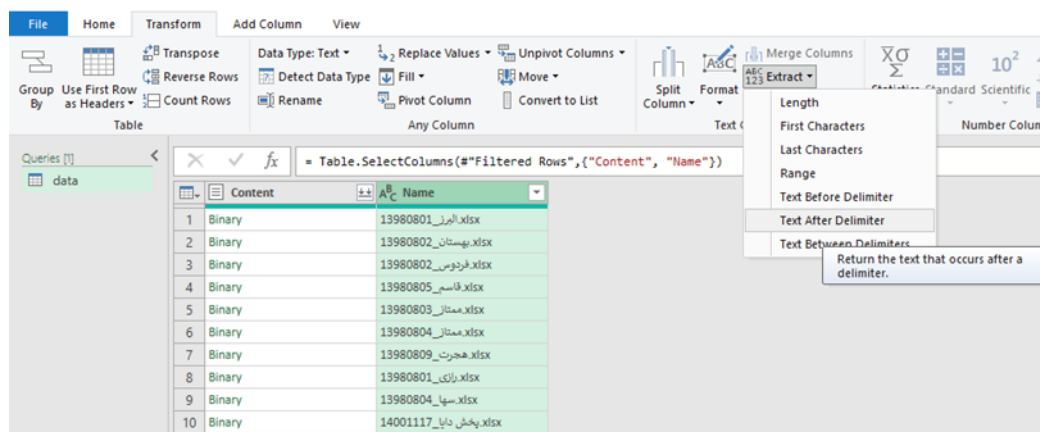
Content Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	
Kind	Excel File
Size	10715
ReadOnly	FALSE
Hidden	FALSE
System	FALSE
Directory	FALSE
Archive	TRUE
Device	FALSE
Normal	FALSE

آنچه برای ما مهم است ستون Content است که باید Extract شود. ستون Name هم مهم است. (در جلسه‌ی قبل دیدیم که در Append کردن، نام شرکت‌ها برای ما مهم بود تا در یک ستون باشند و لازم نباشد به صورت دستی ستون اضافه کنیم و نام شرکت را بنویسیم) اما اینجا خود اسم فایل، همان نام شرکت است. پس در زمان Save کردن فایل باید فایل‌ها را طبق یک الگو نام‌گذاری کرد تا بتوان در این مرحله از نام فایل استفاده کنیم و این ستون جداگانه برای ما لازم است. یک ستون پسوند (Extension) هم هست که فعلاً مهم است. در ستون Extension روی Filter می‌رویم و فقط فایل‌های .xlsx را تیک می‌زنیم. یا می‌توانیم از طریق Text Filter گزینه‌ی Equals را انتخاب کنیم و بگوییم فقط مساوی باشد با همان فایل مدنظر. این گونه فیلتر فعال می‌شود.

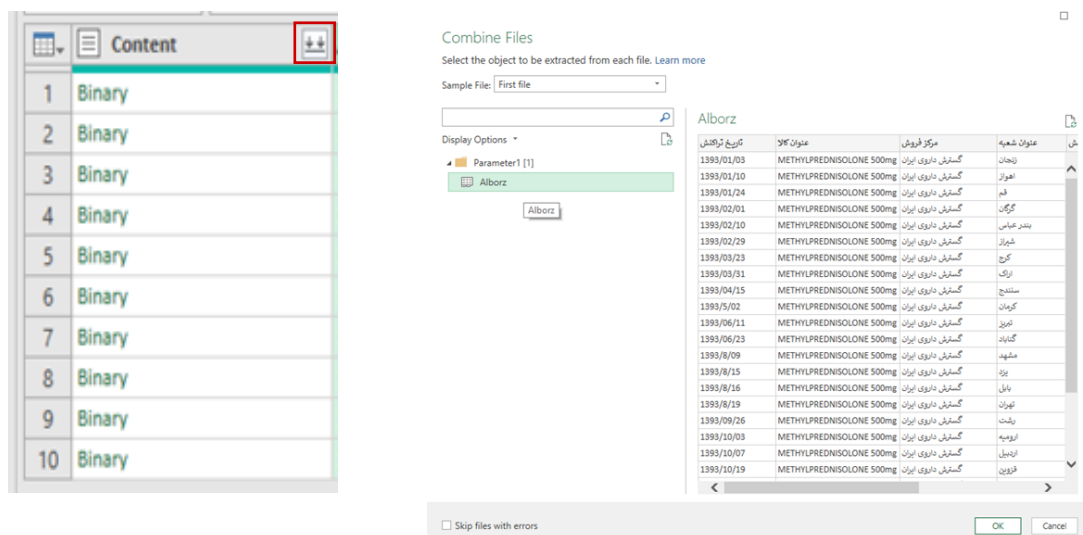


بقیه‌ی ستون‌ها ممکن است اطلاعات اضافی باشند. برای حذف آن‌ها می‌توانیم همه را با Ctrl + Shift بگیریم و یا روی ستونی که می‌خواهیم نگه داریم کلیک کنیم، کلیک راست کرده و Remove Other Columns را بزنیم تا فقط ستون‌های لازم باقی بمانند.

نام ستون‌ها و تاریخ‌ها را هم می‌توانیم تغییر ایجاد کنیم. مثلاً اگر فقط نام‌ها را بخواهیم و تاریخ‌ها برایشان مهم نباشد، در تب Transform از گزینه‌ی Extract > Text Before Delimiter استفاده می‌کنیم تا فقط متنی که قبل از جداکننده (مثلاً Underline) است نگه داشته شود و بقیه حذف شود.



اما در درون هر فایل Excel ممکن است تعداد شیت‌ها متنوع باشد. پس با گزینه‌ی Combine و علامتی که بالای ستون در کنار ستون موجود دارد، کارمان انجام نمی‌شود. در واقع انجام این کار اشتباه است، چون با Combine، پوشه‌ها و پارامترهایی به‌صورت خودکار تشکیل می‌شوند که درست نیستند. پس در این مرحله روی Combine کلیک نمی‌کنیم.

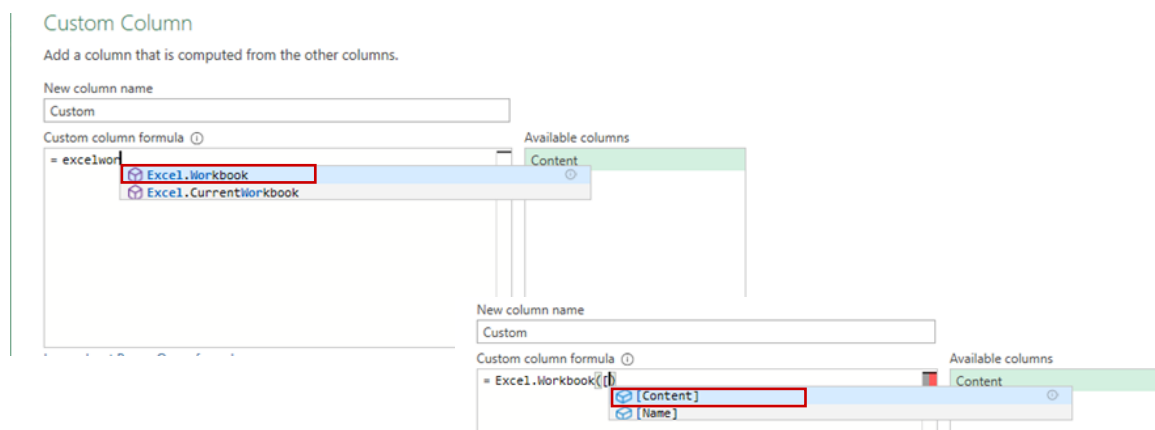


راه‌حل درست این است که باید یک ستون جدید بسازیم که محتویات فایل‌های Excel را برای ما نمایش دهد. برای این کار با کلیک روی Add Column و گزینه‌ی Custom Column را انتخاب می‌کنیم. در کادر فرمول، بعد از علامت مساوی (=)، یک تابع می‌نویسیم. تابع Excel.Workbook را وارد می‌کنیم. کافی است چند حرف اول را بنویسیم تا بقیه‌ی آن نمایش داده شود.

نکته مهم: این زبان به حروف بزرگ و کوچک حساس است. (Case Sensitive)

سپس پرانتز باز و بسته را می‌گذاریم و نام ستونی را که می‌خواهیم صدا بزنی داخل آن می‌نویسیم. در اینجا ما ستون Content را نیاز داریم. پس می‌نویسیم:

= Excel.Workbook([Content])



نکته: باید بدانیم که Excel.Workbook یک تابع است، بنابراین حتماً پرازنتر دارد، و Content نام یک ستون است، پس باید داخل براکت [] نوشته شود. (در جلسه دهم درباره‌ی این تابع به‌طور کامل توضیح داده می‌شود).

با نوشتن این فرمول، یک ستون جدید ساخته می‌شود. حالا دیگر ستون Content را پاک می‌کنیم چون به آن نیازی

نداریم.

	Name	Data	Item	Kind	Hidden
1	13980801_البرز.xlsx	Table	Alborz	Sheet	FALSE
2	13980802_بهبستان.xlsx	Table			
3	13980802_فریدوس.xlsx	Table			
4	13980805_قاسم.xlsx	Table			
5	13980803_ممتاز.xlsx	Table			
6	13980804_ممتاز.xlsx	Table			
7	13980809_هجرت.xlsx	Table			
8	13980801_رازی.xlsx	Table			
9	13980804_سها.xlsx	Table			
10	14001117_پخش دایا.xlsx	Table			

در این ستون جدید، نام تمام Table ها و Sheet ها نشان داده می‌شود. با کلیک روی هر کدام، در پایین صفحه اطلاعات بیشتری ظاهر می‌شود. مثلاً در ستون Kind مشخص است که این مورد یک Table دارد یا Sheet. البته ممکن است همیشه Table نداشته باشد.

در بخش Merge دیده بودیم که ممکن است یک Table در یک سلول هم بصورت فشرده بیاید و حالا هر سلول Table خود حاوی یک سری اطلاعات در قالب جدول است.

در پایین صفحه، موارد مختلفی مانند Name که نام داده یا Sheet مربوط به آن شرکت را نشان می‌دهد، Kind که نوع آن را مشخص می‌کند (مثلاً Sheet، Table یا Defined Name) و Hidden که مشخص می‌کند آیا آن بخش پنهان (Hidden) است یا خیر، وجود دارد. (در VBA، نام Sheet برای برنامه‌نویسی با نامی که کاربر در Excel می‌بیند فرق می‌کند).

بعد از این مرحله، باید داده‌ها را Expand کنیم.

	Name	Data	Kind	Hidden
1	البرز	Table	Sheet	FALSE
2	البرز	Table	DefinedName	TRUE
3	بهبستان	Table	Sheet	FALSE
4	بهبستان	Table	Table	FALSE
5	بهبستان	Table	Table	FALSE
6	فریدوس	Table	Sheet	FALSE
7	فریدوس	Table	Table	FALSE
8	قاسم	Table	Sheet	FALSE
9	قاسم	Table	Table	FALSE
10	ممتاز	Table	Sheet	FALSE
11	ممتاز	Table	Table	FALSE
12	ممتاز	Table	Sheet	FALSE

با کلیک روی علامت فلش کنار ستون، ستون‌های لازم را انتخاب می‌کنیم تا باز شوند. سپس هر ستون را بر اساس نیازمان فیلتر می‌کنیم. مثلاً:

- ستون Hidden را روی مقدار False فیلتر می‌کنیم (یعنی فقط بخش‌های غیرپنهان نمایش داده شوند).
 - ستون Kind را روی مقدار Sheet فیلتر می‌کنیم.
- در نهایت، یک ستون Data باقی می‌ماند که داده‌ها را در قالب Table نگه می‌دارد. وقتی روی هر کدام کلیک کنیم، در پایین صفحه اطلاعات جدول آن فایل نمایش داده می‌شود، مثل همان چیزی که در بخش Merge دیدیم. در اینجا یک ستون داریم که نام شرکت از نام فایل خوانده شده و در مقابل آن ستونی قرار دارد که داده‌های کامل جدول را دارد. این دو ستون بعداً می‌توانند با هم Merge شوند.
- اما یک مشکل وجود دارد: در جدول پایینی، Header یا سرستون‌ها خوانده نشده‌اند.

fx = Table.RemoveColumns("#Filtered Rows2",{"Kind"})

ABC	Name	ABC	Data
1	البرز	Table	
2	بهبستان	Table	
3	فردوس	Table	
4	قاسم	Table	
5	ممتاز	Table	
6	ممتاز	Table	
7	هجرت	Table	
8	رانگی	Table	
9	سها	Table	
10	پخش دایا	Table	

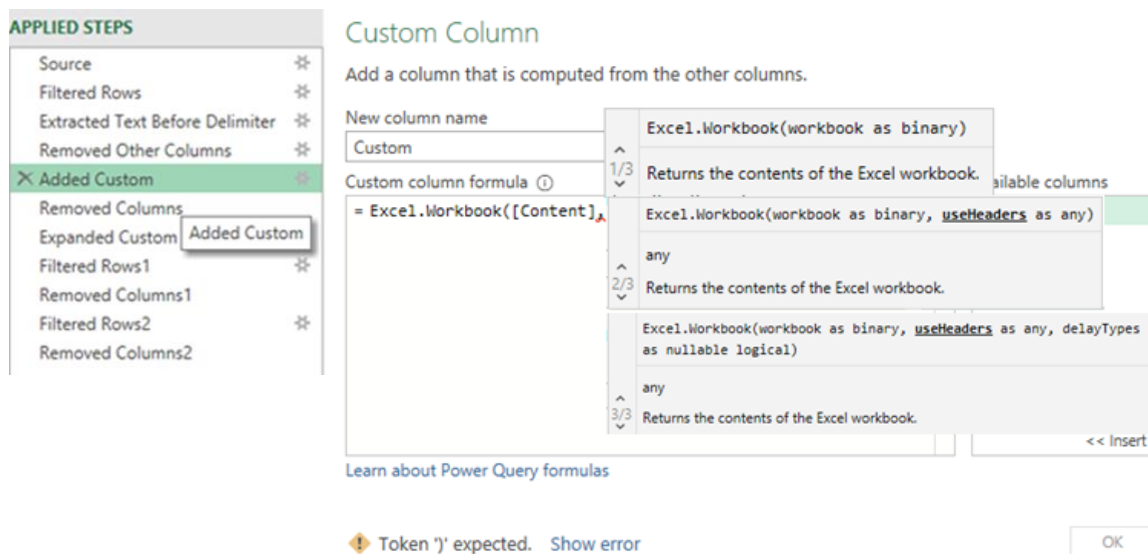
Column1	Column2	Column3	Column4	Column5	Column6	Column7
تاریخ دریافت گزارش	تعداد	موجودی بخش	عنوان شعبه	مرکز فروش	عنوان کالا	تاریخ تراکنش
1393/11/30	0	180	همدان	گسترش داروی ایران	METHYLPREDNISOLONE 500 mg	1393/11/18
1393/11/30	300	830	اصفهان	گسترش داروی ایران	METHYLPREDNISOLONE 500 mg	1393/11/04
1393/10/30	0	130	گرگان	گسترش داروی ایران	METHYLPREDNISOLONE 500 mg	1393/10/26
1393/10/30	720	1110	مشهد	گسترش داروی ایران	METHYLPREDNISOLONE 500 mg	1393/10/12
1393/01/31	0	0	تهران	زیست دارو آوا	Carboplatine 150mg/15ml	1393/01/02
1393/10/30	120	2170	ساری	گسترش داروی ایران	METHYLPREDNISOLONE 500 mg	1393/10/07

دو روش برای انجام این کار داریم:

- استفاده از پارامتر دوم در تابع Excel.Workbook([Content], true)
 - استفاده از تابع Table.PromoteHeaders([Data])
- می‌توانیم در شرایط مختلف از هر کدام استفاده کنیم. برای رفع این مشکل، باید در همان تابعی که نوشتیم، یعنی Excel.Workbook([Content])، یک پارامتر جدید اضافه کنیم. به قسمت Step فرمول مربوطه می‌رویم و بعد از تابع، یک کاما (,) می‌گذاریم. روبروی آن سه حالت ممکن نمایش داده می‌شود:
۱. حالت اول: فقط فایل را به تابع می‌دهیم.

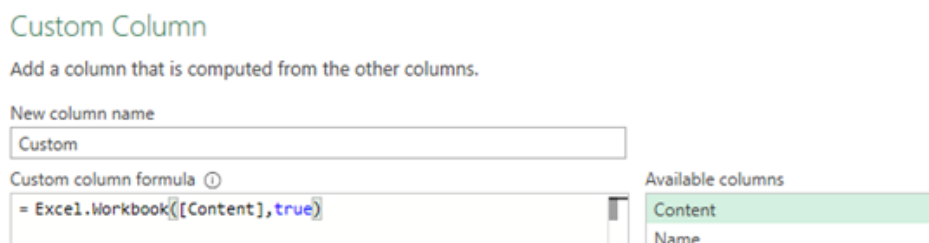
۲. حالت دوم: فایل را می‌دهیم و مشخص می‌کنیم آیا سطر اول را به‌عنوان Header در نظر بگیرد یا نه.

۳. حالت سوم: تنظیمات پیشرفته‌تر.



ما حالت دوم را انتخاب می‌کنیم و مقدار true را می‌نویسیم تا سطر اول به‌عنوان Header در نظر گرفته شود. فرمول نهایی به این صورت خواهد بود:

= Excel.Workbook([Content], true)



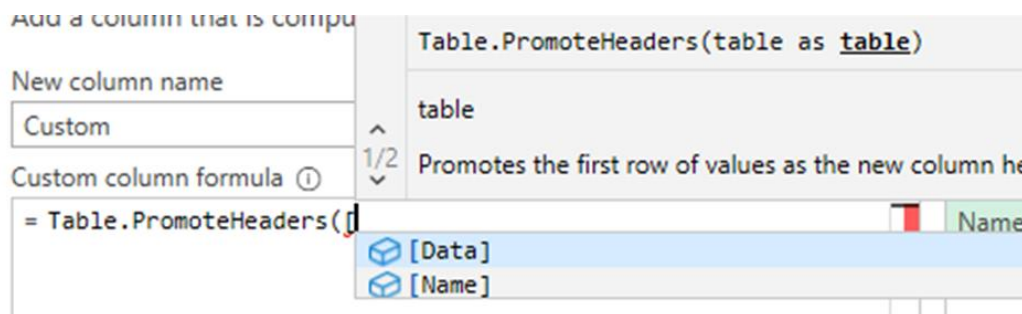
اکنون سرستون‌ها درست نمایش داده می‌شوند و در حالت Expand، همه‌ی سرستون‌ها قابل مشاهده هستند و می‌توانیم تیک هر کدام را بزنیم تا باز شوند.

Table.RemoveColumns("#Filtered Rows2",{"Kind"})						
Name	ABC	123	Data			
1	البرز		Table			
2	بهبستان		Table			
3	فرزادوس		Table			
4	قاسم		Table			
5	ممتاز		Table			
6	عماد		Table			
7	هجرت		Table			
8	زانی		Table			
9	سپا		Table			
10	پخش دایا		Table			

تاریخ دریافت گزارش	تعداد	موجودی پخش	عنوان شعبه	مرکز فروش	عنوان کالا	تاریخ تراکاش
0 1393/01/30	0		زنجان	گسترش داروی ایران	METHYLPREDNISOLONE 500mg	1393/01/03
125 1393/01/30	125		اهواز	گسترش داروی ایران	METHYLPREDNISOLONE 500mg	1393/01/10
0 1393/01/30	59		قم	گسترش داروی ایران	METHYLPREDNISOLONE 500mg	1393/01/24
0 1393/02/31	180		گرگان	گسترش داروی ایران	METHYLPREDNISOLONE 500mg	1393/02/01
5600 1393/02/31	0		بندر عباس	گسترش داروی ایران	METHYLPREDNISOLONE 500mg	1393/02/10

راهکار دوم: می‌خواهیم Header ها (سرستون‌ها) را به سطر بالا ببریم تا تبدیل به عنوان ستون‌ها شوند. برای این کار باید یک Rule یا در واقع یک مرحله‌ی جدید در Power Query تعریف کنیم. به تب Add Column می‌رویم و گزینه‌ی Custom Column را انتخاب می‌کنیم. در کادر فرمول، بعد از علامت مساوی (=)، تابع زیر را می‌نویسیم:

= Table.PromoteHeaders([Data])



در این تابع، پرانتز باز و بسته را می‌گذاریم و نام ستونی را که می‌خواهیم Header های آن به سطر اول منتقل شوند، درون پرانتز قرار می‌دهیم. در اینجا ما ستون Data را می‌خواهیم، چون Header های آن در حال حاضر در ردیف اول داده‌ها هستند. سپس OK را می‌زنیم و مشکل سرستون‌ها حل می‌شود.

بعد از این مرحله، در صورت نیاز می‌توانیم ستون‌هایی را که هنوز Header ندارند یا داده‌های غیرضروری دارند، تیک نزنیم و آن‌ها را در مرحله‌ی Expand فیلتر کنیم.

ردیف	نام	تاریخ تراکشی	عنوان کالا	مرکز فروش	عنوان شعبه	موجودی بخش
1	البرز	1393/01/03	METHYLPREDNISOLONE 500...	گسترش داروی ایران	نجان	
2	البرز	1393/01/10	METHYLPREDNISOLONE 500...	گسترش داروی ایران	اهواز	
3	البرز	1393/01/24	METHYLPREDNISOLONE 500...	گسترش داروی ایران	قم	
4	البرز	1393/02/01	METHYLPREDNISOLONE 500...	گسترش داروی ایران	گرگان	
5	البرز	1393/02/10	METHYLPREDNISOLONE 500...	گسترش داروی ایران	بندر عباس	
6	البرز	1393/02/29	METHYLPREDNISOLONE 500...	گسترش داروی ایران	شیراز	
7	البرز	1393/03/23	METHYLPREDNISOLONE 500...	گسترش داروی ایران	کرج	
8	البرز	1393/03/31	METHYLPREDNISOLONE 500...	گسترش داروی ایران	اراک	
9	البرز	1393/04/15	METHYLPREDNISOLONE 500...	گسترش داروی ایران	سلیمان	
10	البرز	1393/5/02	METHYLPREDNISOLONE 500...	گسترش داروی ایران	کرمان	
11	البرز	1393/06/11	METHYLPREDNISOLONE 500...	گسترش داروی ایران	تهران	
12	البرز	1393/06/23	METHYLPREDNISOLONE 500...	گسترش داروی ایران	گناپاد	
13	البرز	1393/8/09	METHYLPREDNISOLONE 500...	گسترش داروی ایران	مشهد	

در نهایت، وقتی کار Power Query تمام شد، گزینه‌ی Close & Load را می‌زنیم تا داده‌ها وارد محیط Excel شوند. در محیط Excel، هر زمان فایل‌های جدیدی به پوشه اضافه شوند، با Refresh، جدول به‌روزرسانی می‌شود. می‌توانیم برای این کار از کلید میان‌بر Ctrl + Alt + F5 استفاده کنیم.

اگر از Pivot Table استفاده کرده باشیم و بخواهیم داده‌ها را در Pivot به‌روزرسانی کنیم، می‌توانیم از میان‌بر Alt + F5 Refresh استفاده کنیم.



باید توجه داشت که اگر تعداد فایل‌ها زیاد شود (مثلاً بالای ۱۰۰۰ فایل)، سیستم برای Load کردن آن‌ها به قدرت پردازشی بالایی نیاز دارد. همچنین اگر یکی از فایل‌ها را از پوشه حذف کنیم، در نتیجه‌ی نهایی هم آن داده پاک می‌شود Power Query فقط فایل‌هایی را می‌خواند که در آن لحظه داخل پوشه وجود دارند.

می‌توانیم پوشه‌ی CSV را هم به همین روش Import کنیم. کافی است بعد از Transform، ستون‌های اضافی را حذف کنیم.

نکته: حتماً باید در ستون Extension از Text Filter > Equals استفاده کنیم و مقدار آن را برابر با پسوند فایل مدنظر (مثلاً .csv) قرار دهیم تا فقط همان نوع فایل‌ها خوانده شوند.

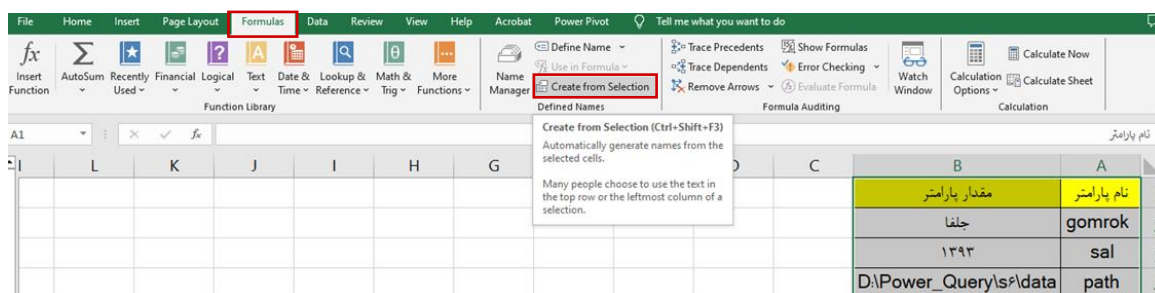
به‌صورت کلی، بهتر است به‌جای Combine خودکار از تابع Excel.Workbook استفاده کنیم تا کنترل بیشتری بر فایل‌ها، Sheet‌ها و داده‌ها داشته باشیم و بتوانیم مراحل را دقیق‌تر مدیریت کنیم.

Parameter و Reduce Rows

این دو کارهای مشابهی انجام می‌دهند. با Parameter چیزهایی را تعریف می‌کنیم که بتوانیم بارها از آنها استفاده کنیم. مثلاً در اینجا Filter کردن را داریم. ابتدا در محیط Excel کار را انجام می‌دهیم و بعد در Power Query .

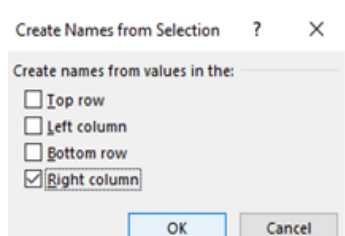
Parameter

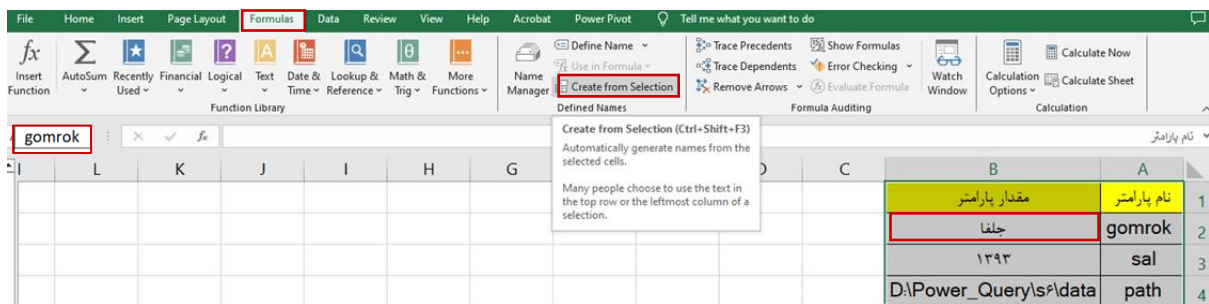
اگر یک ستون در Power Query را بر اساس یک نام Filter کنیم و بعد در Excel ذخیره کنیم، می‌توانیم با تعریف یک جدول دیگر در Excel آن ستون فیلترشده را بر اساس نیاز خودمان Refresh کنیم. برای مثال در شیت Excel جدولی داریم با سرستون‌های «نام پارامتر» و «مقدار پارامتر»، این‌ها قرار است Dynamic باشند. البته بهتر است این شیت را در یک Excel جداگانه ذخیره کنیم.



تکنیکی که در اکسل استفاده می‌شود اینطور است که اگر بخواهیم سلول‌ها نام پارامترها را به خودشان بگیرند، محدوده داده‌ها را انتخاب می‌کنیم، بعد با کلیدهای میانبر **Ctrl + Shift + F3** یا از تب **Formulas** گزینه **Create from Selection** را انتخاب می‌کنیم.

اگر نام‌ها در سمت راست باشند، تیک **Right Column** را می‌زنیم و **OK** می‌کنیم. حالا اگر در جدول، مثلاً روی سلول "جلفا" کلیک کنیم، در **Name Box** کلمه "گمرک" نوشته می‌شود و همین‌طور برای سطرهای بعدی. در واقع سلول‌ها انگار نام گرفته‌اند.





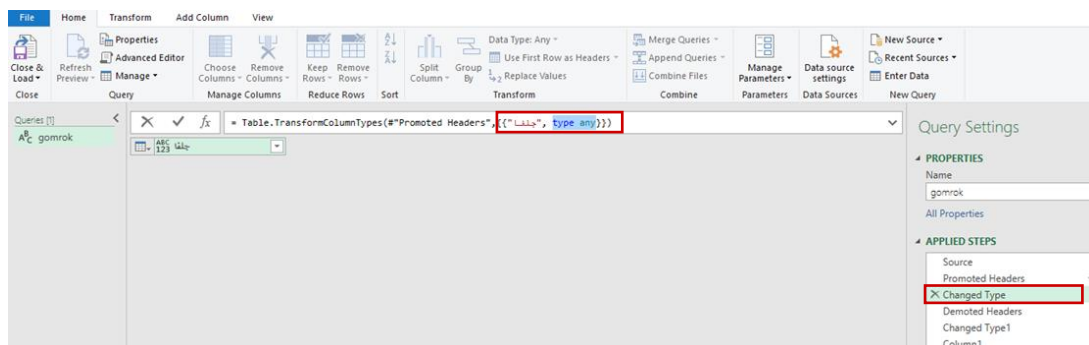
حالا اگر روی یکی از سلول‌ها باشیم و از قسمت Data → From Table/Range دیگر صفحه‌ای نمایش داده نمی‌شود که آیا می‌خواهید Convert to Table انجام دهید یا خیر، بلکه مستقیماً وارد محیط Power Query می‌شود و یک سلول به نام مثلاً "جلفا" را نشان می‌دهد.

در واقع نام سلولی که در اکسل روی آن کلیک کرده بودیم. اما چون در Power Query همه دیتاها باید در قالب جدول و دارای Header باشد، این کلمه "جلفا" به عنوان Header می‌آید. برای اینکه از حالت Header بیرون بیاییم، از گزینه Drill Down استفاده می‌کنیم تا مقدار آن کلمه را داشته باشیم.



از این به بعد، Query گمرک = مقدار جلفا خواهد بود.

نکته: در Steps باید Changed Type ها را حذف کنیم.



حالا در Query اول که داشتیم (اطلاعات تمام خودروهای وارداتی)، در قسمت Hardcode برای فیلتر اولی، مثلاً یک شهر را فیلتر کرده بودیم. اسم شهر را برمی‌داریم و به جای آن، کلمه "گمرک" را به عنوان یک Query می‌نویسیم.

Queries [2] × ✓ f_x = Table.SelectRows("#Changed Type1", each ([گمرک] = "قشم"))

کشور طرف معامله	گمرک	ماه	سال
امارات متحده عربي	قشم	1	1394
امارات متحده عربي	قشم	1	1394
چين	قشم	2	1394
امارات متحده عربي	قشم	2	1394
امارات متحده عربي	قشم	3	1394
امارات متحده عربي	قشم	6	1394
امارات متحده عربي	قشم	6	1394

× ✓ f_x = Table.SelectRows("#Changed Type1", each ([گمرک] = gomrok))

کشور طرف معامله	گمرک	ماه	سال
امارات متحده عربي	قشم	1	1394
امارات متحده عربي	قشم	1	1394
چين	قشم	2	1394
امارات متحده عربي	قشم	2	1394

gomrok
Geometry.FromWellKnownText
Geometry.ToWellKnownText
GeometryPoint.From
Geography.FromWellKnownText
GeographyPoint.From

در این صورت جدول در ستون گمرک، "جلفا" را به عنوان مقصد نشان می دهد.

Queries [2] × ✓ f_x = Table.SelectRows("#Changed Type1", each ([گمرک] = gomrok))

کشور طرف معامله	گمرک	ماه	سال
امارات متحده عربي	جلفا	1	1394
جمهوري كره	جلفا	1	1394
آلمان	جلفا	2	1394
امارات متحده عربي	جلفا	2	1394
امارات متحده عربي	جلفا	3	1394
امارات متحده عربي	جلفا	3	1394
امارات متحده عربي	جلفا	3	1394
امارات متحده عربي	جلفا	4	1394
امارات متحده عربي	جلفا	4	1394
امارات متحده عربي	جلفا	4	1394
امارات متحده عربي	جلفا	4	1394
امارات متحده عربي	جلفا	4	1394
امارات متحده عربي	جلفا	4	1394
امارات متحده عربي	جلفا	5	1394

پس اگر در شیت اکسل، مقدار شهر را تغییر دهیم — مثلاً "جلفا" بشود "یزد" — با فشردن کلیدهای Alt + Ctrl

Refresh می شود، داده ها را می خواند و در جدول فقط واردات "یزد" را نمایش می دهد.

نام پارامتر	مقدار پارامتر
gomrok	یزد
sal	۱۳۹۳
path	D:\Power_Query\ls\data

سال	ماه	گمرک	کشور طرف معامله	تعرفه	نام کالا
1394	8	پزد	امارات متحده عربی	87032319	یک دستگاه خودروی نو 28i XDRIVE
1394	8	پزد	امارات متحده عربی	87032329	یک دستگاه سواری نوهیوندا سوناتا
1394	8	پزد	امارات متحده عربی	87032329	یک دستگاه سواری نوکیاسپورتج 4 سی
1394	8	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری نوکیا اینیما 4
1394	8	پزد	امارات متحده عربی	87032329	یک دستگاه سواری هیوندا توسان 5
1394	8	پزد	امارات متحده عربی	87032329	یک دستگاه سواری هیوندا سوناتا ال اف
1394	10	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری نوکیا اینیما
1394	10	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	10	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	10	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا توسان
1394	10	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	10	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	11	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	11	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	11	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س
1394	11	پزد	امارات متحده عربی	87032329	یک دستگاه خودرو سواری هیوندا س

به این ترتیب، در واقع Parameter ساخته می‌شود. شیت‌ها را می‌توان Hide هم کرد و Query ها همچنان Refresh می‌شوند.

حالا باید پارامتر را در محیط Power Query بسازیم. از قسمت Manage Parameters وارد می‌شویم. در بخش Name مثلاً "سال" را وارد می‌کنیم. در بخش Type، گزینه‌ی Decimal Number را انتخاب می‌کنیم. و در قسمت Current Value مثلاً عدد ۱۳۹۳ را قرار می‌دهیم. به این ترتیب، در بین Queries یک Query جدید به نام «سال» ساخته می‌شود که مقدار ۱۳۹۳ را نگه می‌دارد.

یک نکته مهم این است که بهتر است نام‌گذاری پارامترها با p_ آغاز شود، تا بدانیم این‌ها پارامترهای سفارشی هستند که ساخته‌ایم.

New

123 p_sal

X

Name

p_sal

Description

☒ Required

Type

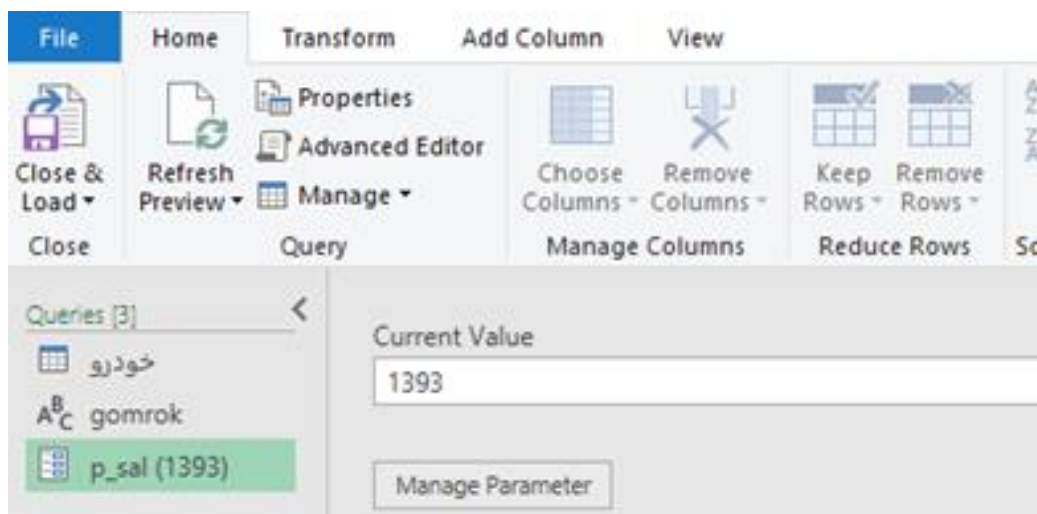
Decimal Number

Suggested Values

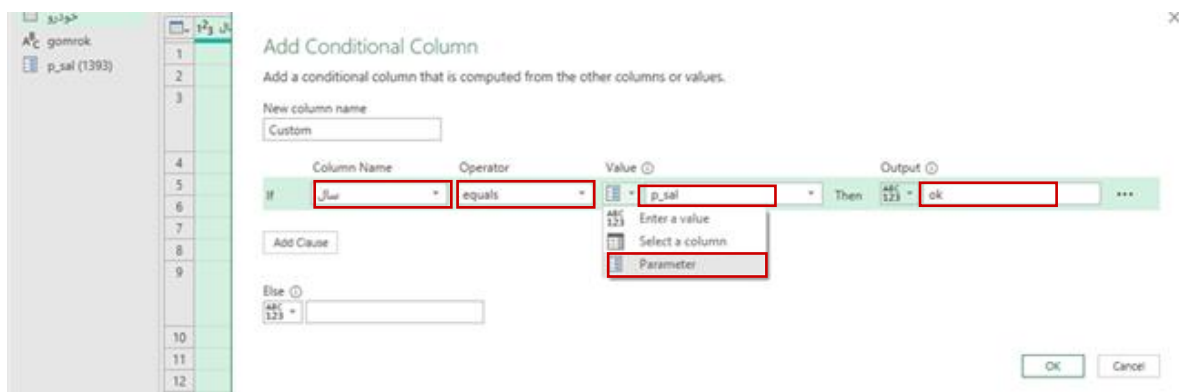
Any value

Current Value

1393



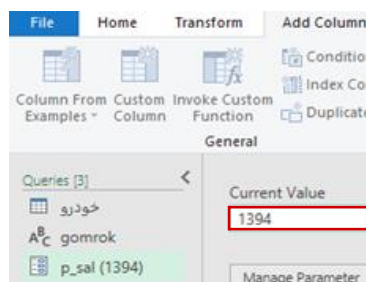
حالا در Query اصلی، از مسیر Add Column → Conditional Column استفاده می کنیم.



یعنی اگر مقدار ستون برابر با ۱۳۹۳ بود، مقدار "OK" در ستون جدید نمایش داده شود، و اگر نبود، مقدار null نمایش داده می شود.

ردیف	سال	گمرک	کشور طرف معامله	Custom
1	1394	گمرک 1	امارات متحده عربی	null
2	1394	گمرک 1	امارات متحده عربی	null
3	1394	گمرک 1	امارات متحده عربی	null
4	1394	گمرک 1	امارات متحده عربی	null
5	1394	گمرک 1	امارات متحده عربی	null
6	1394	گمرک 1	امارات متحده عربی	null
7	1394	گمرک 1	امارات متحده عربی	null
8	1394	گمرک 1	امارات متحده عربی	null
9	1394	گمرک 1	امارات متحده عربی	null

حالا اگر در Query سال، مقدار را از ۱۳۹۳ به ۱۳۹۴ تغییر دهیم، در Query اصلی همه ی واردات مربوط به سال ۱۳۹۴ نمایش داده می شوند، و در ستون جدید فقط سطرهایی که شرط برقرار است مقدار OK خواهند داشت.



ردیف	سال و ق	ماه و ق	گمرک	کشور طرف معامله	Custom
1	1394	1	گناوه	امارات متحده عربي	ok
2	1394	1	گناوه	امارات متحده عربي	ok
3	1394	1	گناوه	امارات متحده عربي	ok
4	1394	1	گناوه	امارات متحده عربي	ok
5	1394	1	گناوه	امارات متحده عربي	ok
6	1394	1	گناوه	امارات متحده عربي	ok
7	1394	1	گناوه	امارات متحده عربي	ok

به این ترتیب، این پارامتر قابلیت فراخوانی (Call) دارد و حتی در UI نیز خوانده می‌شود.

وقتی پارامتر در Excel ساخته می‌شود (یا از هر منبع داده‌ای خوانده شود)، می‌تواند در Queries، Code‌های مختلف استفاده شود. اما وقتی پارامتر را مستقیماً از بخش Manage Parameters بسازیم، در خود محیط Power Query UI هم می‌توانیم از آن استفاده کنیم. همچنین بهتر است برای پارامترها Suggested Values تعریف کنیم، مثلاً لیستی از اسامی شهرهای پرکاربرد برای واردات در مثال بالا.

بعد از ساخت پارامتر، در Query اصلی می‌توانیم در ستون "گمرک" شهرها را فیلتر کرد. در بخش Formula Bar هم مقدار فیلترشده را با نام پارامتر جایگزین می‌کنیم. بر اساس پارامتر تعریف‌شده، جدول فقط همان شهر را نمایش می‌دهد. اگر مقدار پارامتر را تغییر دهیم، شهرها در جدول اصلی بر اساس آن پارامتر فیلتر می‌شوند. به این ترتیب، می‌توانیم داده‌ها را از یک نقطه مرکزی مدیریت و فیلتر کنیم، و هر جا که نیاز به یک مقدار ثابت داریم، از Parameter استفاده کنیم.

نکته: تغییر نام (Rename) پارامترها هیچ تأثیری بر Code‌ها ندارد، چون در همان لحظه همه‌ی نام‌ها در کل Query به‌روزرسانی می‌شوند.

پس بطور دو روش برای ساخت پارامتر وجود دارد:

۱. از طریق Manage Parameters در Power Query.

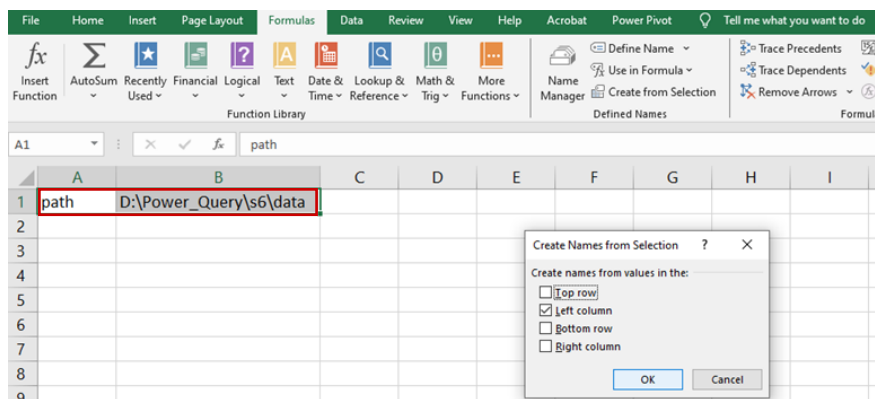
۲. از طریق سلول‌های Excel و با تعریف پویا (Define Name).

تحلیل و بررسی داده

می‌توان در Excel مسیر فایل را تعریف کرد و آن را به Query آورد. در Advanced Editor کوئری اصلی،

محل‌هایی که مسیر فایل به‌صورت متن صریح نوشته شده را با نام متغیری که در یک Query دیگر تعریف کرده‌ایم جایگزین

می کنیم و تغییرات را تأیید می کنیم.



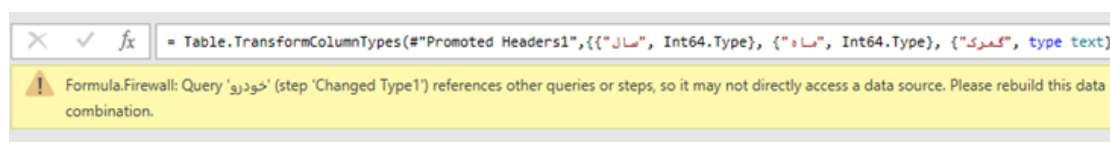
```
let
Source = Excel.Workbook(File.Contents("D:\Power_Query\s6\data\khodro.xlsx"), null, true),
خودرو_Sheet = Source[Item="خودرو",Kind="Sheet"][Data],
#"Promoted Headers" = Table.PromoteHeaders(خودرو_Sheet, [PromoteAllScalars=true]),
#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"1394 ماهه سال 12", {Type="Text"}}}),
#"Promoted Headers1" = Table.PromoteHeaders(#"Changed Type", [PromoteAllScalars=true]),
#"Changed Type1" = Table.TransformColumnTypes(#"Promoted Headers1",{{"سال", Int64.Type}, {"ماه", Int64.Type}, {"تغییر", Int64.Type}}),
in
#"Changed Type1"
```

```
let
    Source = Excel.Workbook(File.Contents(path & "\khodro.xlsx"), null, true),
    خودرو_Sheet = Source{[Item="خودرو", Kind="Sheet"]}[Data],
```

ممکن است هنگام اجرا با خطای مواجه شویم که معمولاً Error مربوط به Formula Firewall است و مشکل

دسترسی است، چون مسیر فایل را به صورت متنی (از سلول‌ها) به کد داده‌ایم و برای دسترسی به آن باید دسترسی‌های لازم در

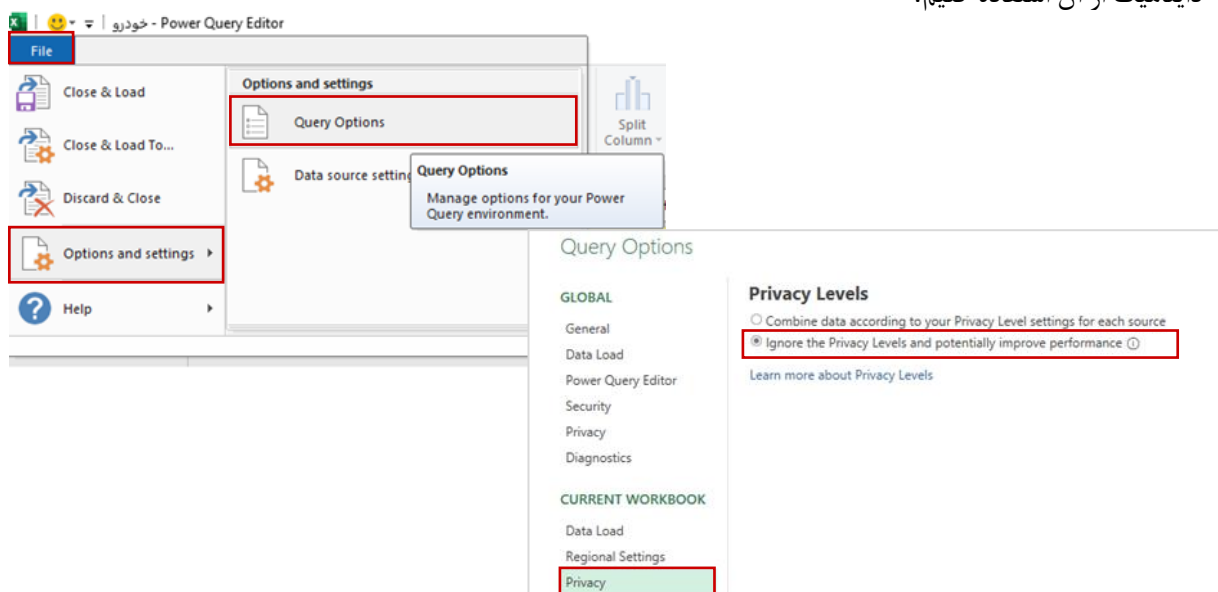
ویندوز/محیط داده وجود داشته باشد.



در این حالت می‌توانیم گزینه Privacy را غیر فعال کنیم: از مسیر

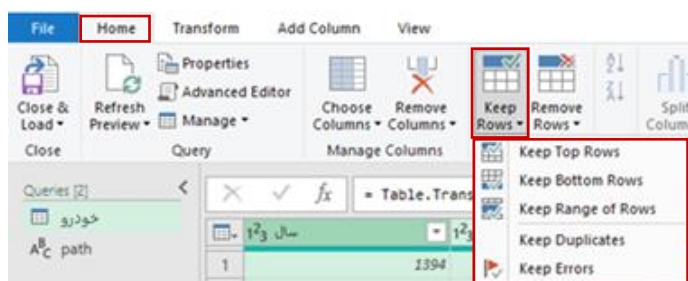
File → Options and Settings → Query Options

تیک Ignore the privacy را می‌زنیم. با این کار می‌توانیم مسیره‌ی فایل را در اختیار کاربران قرار دهیم و به صورت داینامیک از آن استفاده کنیم.



Reduce Rows

این بخش شامل Keep Rows و Remove Rows می‌شود. انواع گزینه‌های Keep Rows عبارتند از:



- Keep Top Rows : سطرهای بالای جدول را نگه می‌دارد.

- Keep Bottom Rows : سطرهای پایین را نگه می‌دارد.

- Keep Range of Rows : یک بازه مشخص از سطرها را نگه می‌دارد.

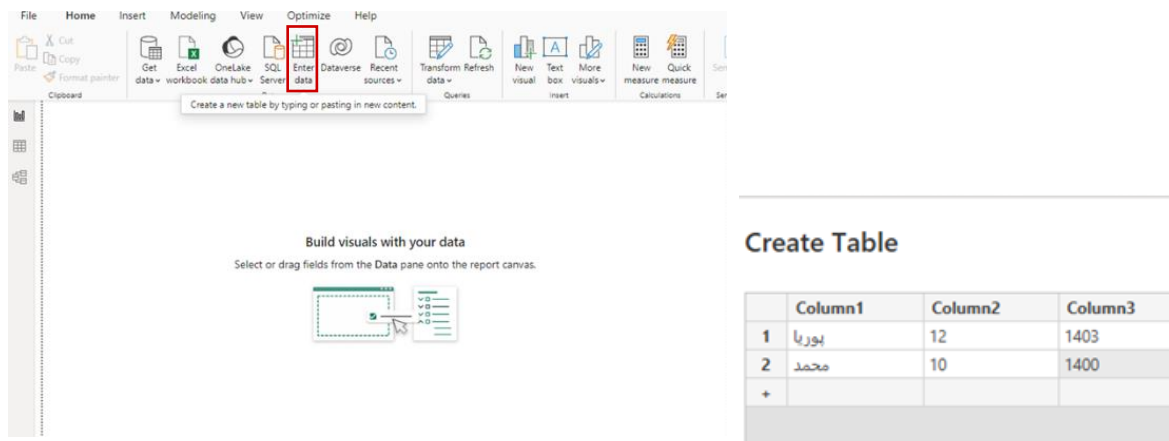
گزینه Keep Duplicate Rows سطرهای تکراری را نگه می‌دارد؛ این در فایل‌های بسیار گسترده مفید است تا سطرهای تکراری را به سرعت پیدا کنیم. باید دقت کرد که برای این کار بهتر است همه ستون‌ها (تمام سلول‌ها) را انتخاب کنی تا هیچ سطر از قلم نیفتد و تشخیص تکراری‌ها صحیح باشد.

گزینه Keep Errors باقی‌مانده سطرهایی که دارای خطا هستند را جدا نگه می‌دارد تا ما بتوانیم متوجه خطاها شویم و آن‌ها را بررسی کنیم.

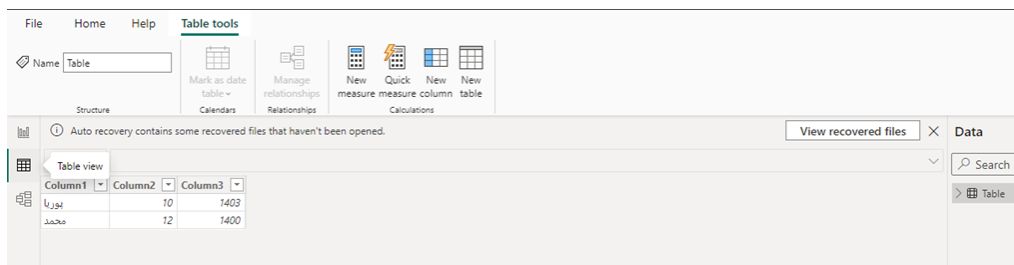
Power Query / Power BI در Remove Rows

گزینه‌ی Remove Rows برای حذف سطرها در Power Query استفاده می‌شود. این گزینه می‌تواند سطرهای بالا یا پایین جدول را حذف کند.

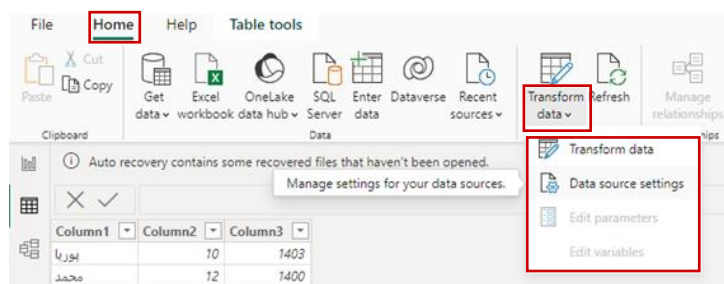
در محیط Power BI می‌توانیم داده‌ای را به‌صورت دستی وارد کنیم: از مسیر Home → Enter Data یک جدول با سطرها و ستون‌های دلخواه می‌سازیم.



وقتی Load می‌شود، این داده مستقیماً در خود Power BI ذخیره (Embed) می‌شود و دیگر فایل خارجی ندارد.



اگر بخواهیم این داده را ویرایش کنیم، باید دوباره وارد محیط Power Query شویم. در تب Home از بخش Transform Data می‌توان جدول را ویرایش کرد.



اما اگر به بخش Advanced Editor برویم، می‌بینیم که کدها به زبان M نوشته شده‌اند و معمولاً به صورت-JSON مانند هستند (یعنی ساختار داده‌ای رمزگونه دارند). ویرایش دستی این کدها کمی سخت است.

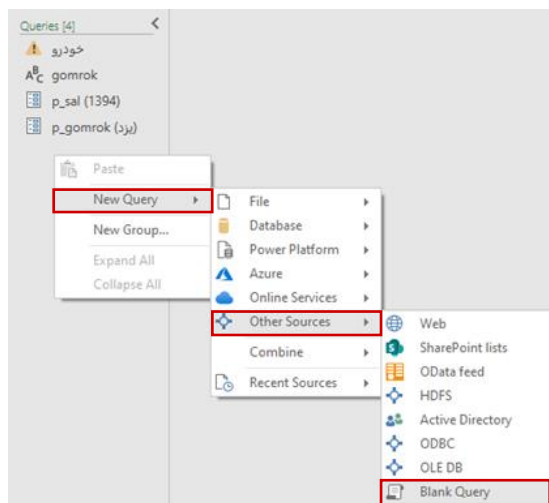


اگر بخواهیم داده را به صورت کد M بنویسیم، می‌توانیم یک متن code M را در جایی بنویسیم یا کپی کنیم و سپس در Advanced Editor آن را جایگذاری (Paste) کنیم. یک نمونه ساختار کلی کد M برای ساخت جدول مانند تصویر زیر است.

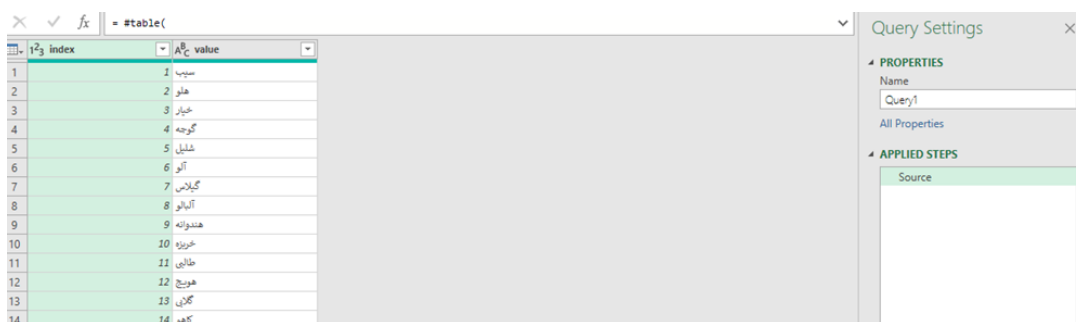
Query1

```
let
Source = #table(
    type table[index = Int32.Type, value = text],
    {
        {1, \"سیب\"},
        {2, \"هلو\"},
        {3, \"کیار\"},
        {4, \"گوجه\"},
        {5, \"تفیل\"},
        {6, \"آلو\"},
        {7, \"گیلان\"},
        {8, \"آلبالو\"},
        {9, \"هندوانه\"},
        {10, \"خربزه\"},
        {11, \"طالبی\"},
        {12, \"هویج\"},
        {13, \"کدایی\"},
        {14, \"کاهو\"}
    }
)
in
Source
```

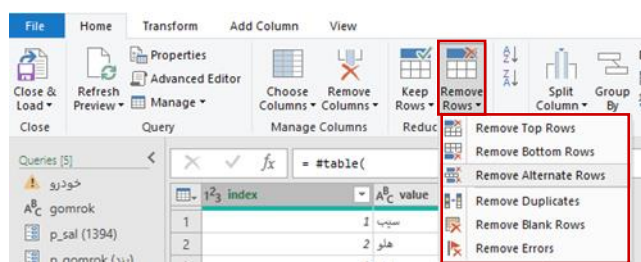
در این ساختار: در بخش #table نوع جدول را تعریف می‌کنیم (ستون‌ها و نوع داده‌ی هر کدام). در { ... } لیستی از سطرها (Sub List) داریم که مقادیر را شامل می‌شود. در پایان in Source برای برگرداندن جدول استفاده می‌شود. برای انجام این کار، در Power Query یک Blank Query جدید ایجاد کنید.



روی آن راست کلیک کرده و **Advanced Editor** را باز کنید. کد بالا را جای گذاری کنید و **Done** را بزنید. در نتیجه، یک جدول جدید ساخته می شود. این جدول قابلیت ویرایش نیز دارد.



Remove Rows



گزینه **Remove Alternate Rows** به شما اجازه می دهد یک الگو برای حذف سطرها تعریف کنید. در این حالت،

باید مشخص کنید:

Remove Alternate Rows

Specify the pattern of rows to remove and keep.

First row to remove

1.2 1

Number of rows to remove

1.2 1

Number of rows to keep

1.2 2

۱. از کدام سطر شروع شود (مثلاً از سطر ۱)،

۲. چند سطر حذف شود، (مثلاً ۱ سطر)

۳. چند سطر نگه داشته شود. (مثلاً ۲ سطر)

fx = Table.AlternateRows(Source,0,1,2)

123 index	ABC value
1	هلو
2	خیار
3	شلیل
4	آلو
5	آلبالو
6	هندوانه
7	طالبی
8	هویج
9	کاهو
10	موز

fx = Table.AlternateRows(Source,0,1,2)

نکته‌ی مهم تفاوت Index در فرمول نویسی و UI این است که در فرمول نویسی اولین سطر اندیس (Index) از 0

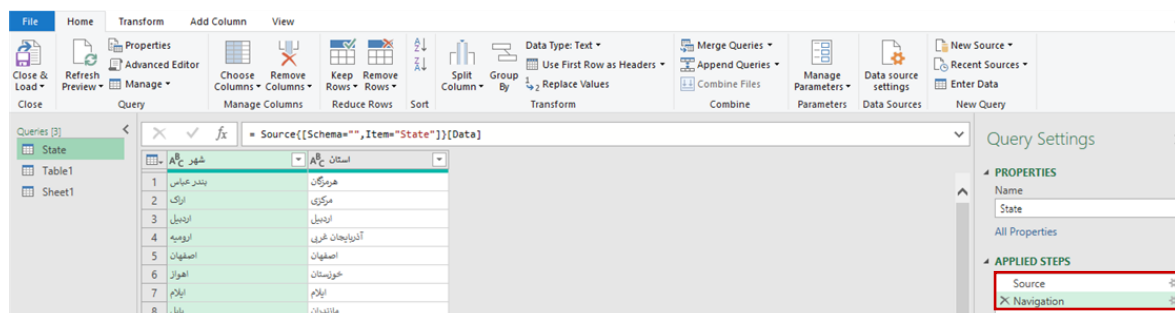
شروع می‌شود، اما در رابط کاربری (UI) اولین سطر به‌صورت 1 نمایش داده می‌شود. بنابراین:

- در UI اگر نوشتید "از سطر ۱ شروع شود"،
 - در کد M در واقع مقدار 0 در نظر گرفته می‌شود.
- پس هنگام نوشتن فرمول دستی، باید دقت کنید که تفاوت بین اندیس صفر و یک را در نظر بگیرید.
- Remove Blank Rows: همه‌ی سطرهای خالی را حذف می‌کند.
 - Remove Errors: سطرهایی را که در داده‌ها خطا (Error) دارند حذف می‌کند.

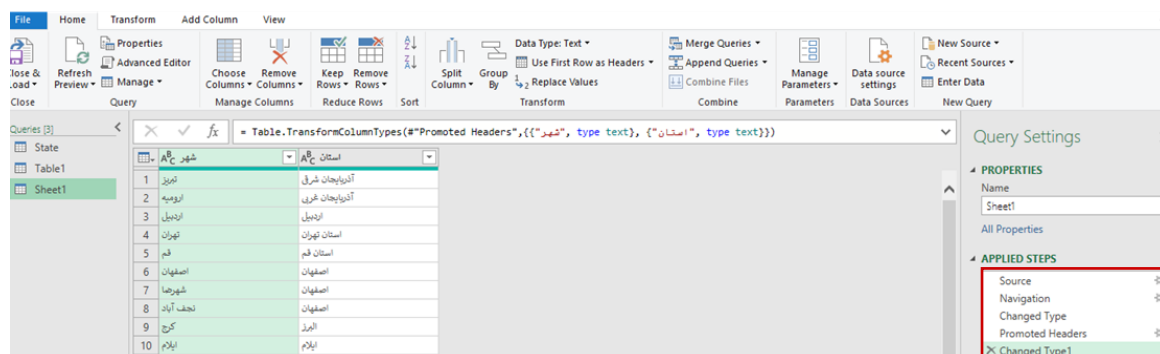
جلسه هفتم

Table، Reference و Duplicate

آوردن جدول‌ها از Excel و یا Access در Power Query ممکن است تفاوت‌هایی داشته باشد. منابع داده ساختارهای مختلفی دارند. مثلاً در Access همه چیز ساختاریافته است، زیرا Database است و اساساً پایگاه داده محسوب می‌شود. اما در Excel لزوماً این طور نیست و ممکن است داده‌ها به شکل‌های مختلفی در سلول‌ها آورده شوند. در نمونه زیر داده از اکسس Import شده است.

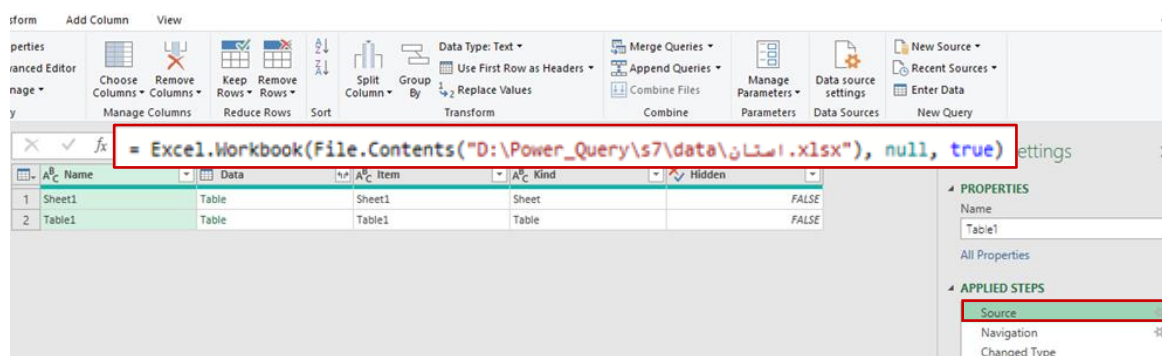


اما در این نمونه داده از اکسل آمده است. (به گام‌های اعمال شده دقت کنید)

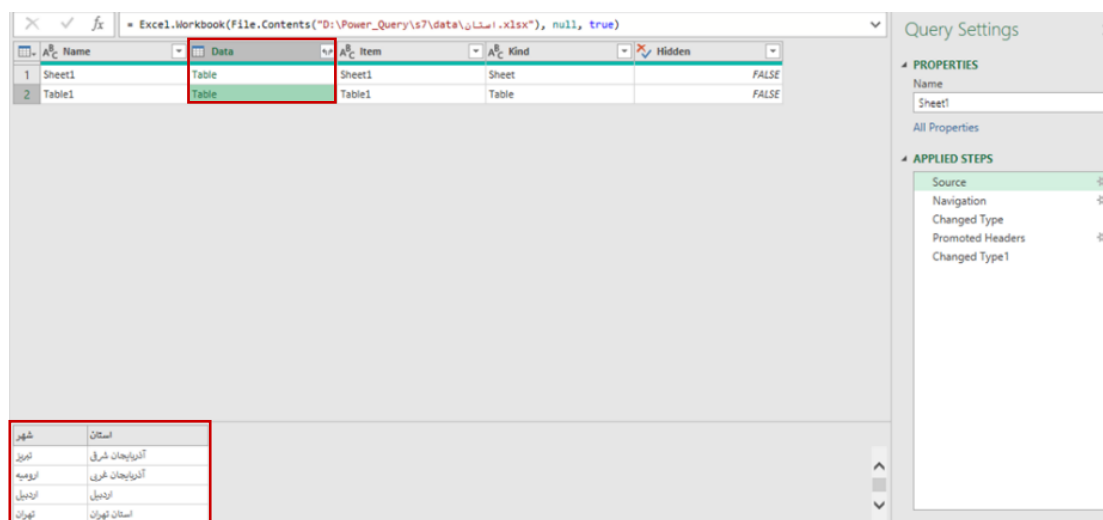


Table

اولین Step در Power Query ها Source است، که اگر یک فایل را از Excel یا جای دیگری خوانده باشد، در Formula Bar نشان می‌دهد که این Query با فلان آدرس، مسیر و نام به یک منبع داده وصل شده است.



اما همیشه به یک منبع داده وصل نیست، چون ممکن است همان جا یک Query ایجاد کرده باشیم. در واقع Source نشان می‌دهد که اصل داده از کجا آمده و ستون‌هایی که نمایش می‌دهد، اطلاعات کلی آن Query را دارند. مثلاً نوع داده (Kind) و اینکه Item ها چه هستند؛ مثلاً داده‌ها از نوع Table است و اطلاعات این‌چنینی است. خروجی Source یک جدول از اطلاعات ساختاری آن فایل است که می‌تواند در Table ها باز هم اطلاعاتی داشته باشد. (این مورد در جلسه Merge و Import Folder تدریس شد) در واقع Table درون Table دیگر باشد.



در سایت PowerQuery.how در بخش Data Types می‌بینیم که جدول انواع داده‌ها را نشان می‌دهد. برای هر Step که اجرا می‌شود، یک Code در پشت‌صحنه نوشته می‌شود که در Advanced Editor قابل مشاهده است. اگر آن را باز کنیم، کدها در هر مرحله دیده می‌شوند. اولین Step همان Source است که برابر است با آدرس فایل و مسیر و مشخصات فایل. دستور بعدی اشاره دارد به دستور قبلی (که اینجا Source بوده است) و دستور بعدی هم به خط قبل خود اشاره دارد. یعنی در هر Step که پیش می‌رویم، در کدنویسی، مرحله قبل آورده می‌شود و در یک Table جدید ریخته می‌شود.

برخی Step ها نام‌هایی دارند که بین کلماتشان فاصله است؛ پس در کدنویسی آن‌ها قبل از اسم Step یک علامت # آورده می‌شود. و نام Step هم در Double Quotation (") قرار می‌گیرد. اما اگر بین کلمات فاصله نباشد، علامت # هم آورده نمی‌شود. برای مثال "Changed Type" و "Sheet1_Sheet"

```
let
    Source = Excel.Workbook(File.Contents("D:\Power_Query\s7\data\استان.xlsx"), null, true),
    Sheet1_Sheet = Source[[Item="Sheet1", Kind="Sheet"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(Sheet1_Sheet,{{"Column1", type text}, {"Column2", type text}}),
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type", [PromoteAllScalars=true]),
    #"Changed Type1" = Table.TransformColumnTypes(#"Promoted Headers",{{"شهر", type text}, {"استان", type text}})
in
    #"Changed Type1"
```

در واقع Let به معنای شروع است و in پایان و خروجی است.

باز در اینجا به سایت PowerQuery.how می‌رویم. تابع Table.PromoteHeaders را سرچ می‌کنیم و جداولش را می‌بینیم.

گاهی می‌خواهیم پاک‌سازی‌هایی که در یک Query انجام داده‌ایم را در یک Query دیگر هم داشته باشیم. در واقع یک کپی یا تکرار از یک Query می‌خواهیم. سه راهکار برای این کار وجود دارد:

۱. اینکه از آن Query، کدهایش را Copy کنیم. یک Query جدید بسازیم. مثلاً از Blank Query و کدها را در بخش Advanced Editor Paste آن کنیم. اما این کار همیشه لازم نیست.

```
let
    Source = Excel.Workbook(File.Contents("D:\Power_Query\data\فروشگاه.xlsx"), null, true),
    Orders_Sheet = Source[[Item="Orders", Kind="Sheet"]][Data],
    #"Promoted Headers" = Table.PromoteHeaders(Orders_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"شماره سفارش", type text}, {"تاریخ سفارش", type date}, {"تاریخ ارسال", type text}}),
in
    #"Changed Type"
```

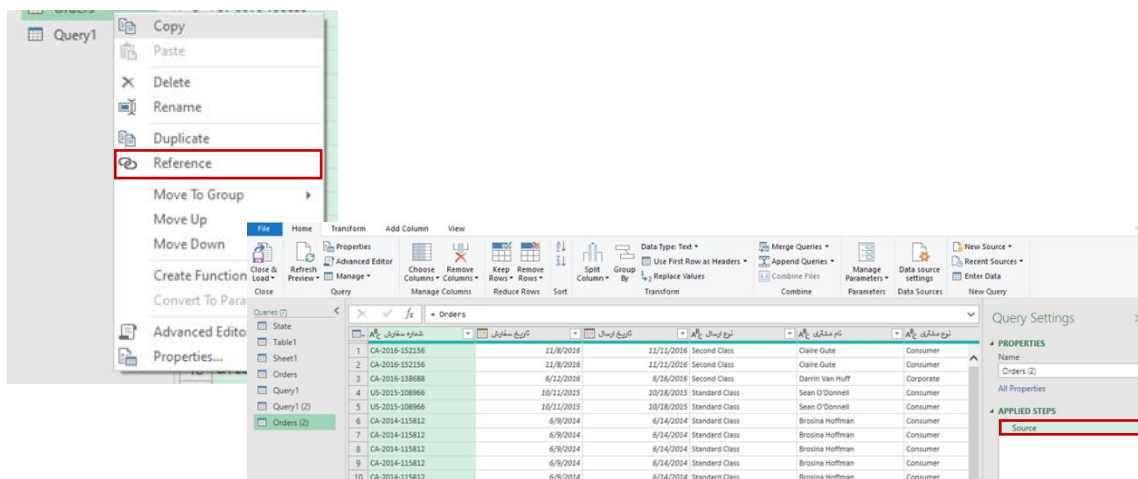
۲. راه دوم این است که همان‌جا روی Query کلیک راست کنیم، Copy بگیریم، و هر جایی که بخواهیم در فایل دیگر آن Query را اضافه کنیم. این روش وقتی مفید است که بخواهیم در فایل دیگری از همان Query و پاک‌سازی‌هایی که انجام داده‌ایم استفاده کنیم.

۳. راه سوم این است که همان‌جا کلیک راست کرده و گزینه Duplicate را بزنیم. همان لحظه یک Query جدید در همان فایل ساخته می‌شود. حالا هر Query برای خودش مستقل است و اگر Step جدیدی در یکی از آن‌ها انجام دهیم، تأثیری روی Query‌های دیگر ندارد.

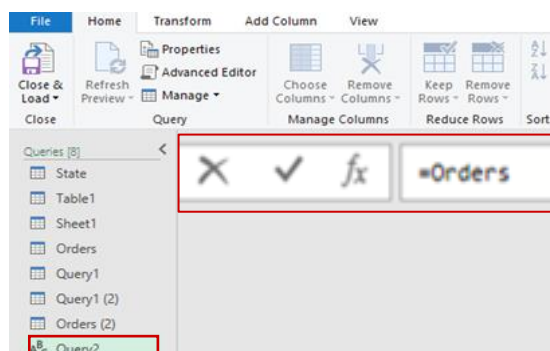
Reference

گاهی روی یک Query کارهای مختلفی انجام داده‌ایم و Step‌های متعددی داشته و سنگین شده است. اما می‌خواهیم بر اساس همان Query اولیه، یک Query جدید بسازیم تا اقدامات متفاوتی روی آن انجام دهیم. در این حالت می‌خواهیم از یکی از Step‌ها یک شاخه (Branch) ایجاد کنیم.

با کلیک راست روی Query و انتخاب گزینه Reference، یک Reference از آن ساخته می‌شود.

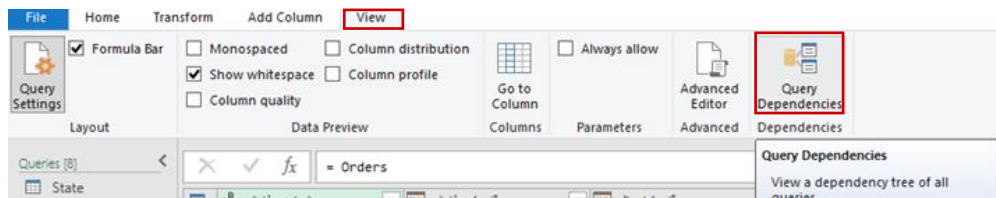


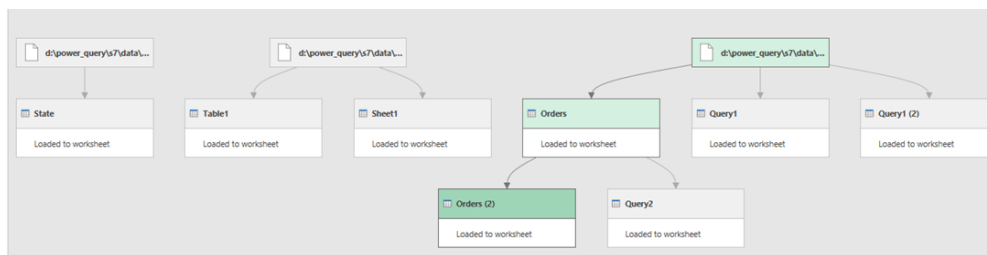
در واقع مانند آن است که یک کپی از آن گرفته‌ایم، اما این Query جدید فقط یک Step دارد، که آن هم Source است. فرمولش برابر با نام Query اصلی است و می‌توانیم حتی یک New Query بسازیم و در فرمول آن، نام Query مورد نظر را بنویسیم. در این صورت هم یک Reference داریم.



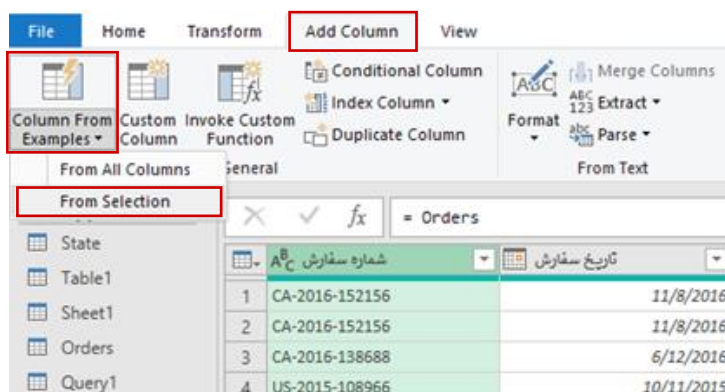
حالا هر اقدامی که روی Query اول انجام داده‌ایم، روی Query جدید نیز اعمال می‌شود، چون فرمول Query جدید مستقیماً از Query اول می‌خواند. اما اگر در Query جدید Step جدیدی اضافه کنیم، هیچ تأثیری روی Query اصلی یا سایر Query هایی که از آن Reference گرفته شده‌اند ندارد.

در تب View بخشی به نام Query Dependencies داریم که وابستگی Query ها به یکدیگر را نشان می‌دهد و می‌توانیم از آن برای مشاهده ارتباطها استفاده کنیم.

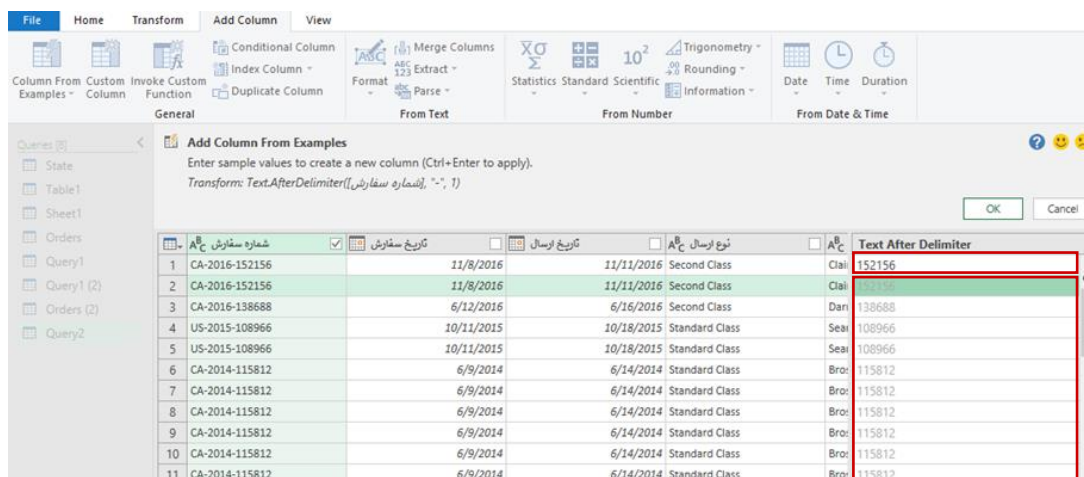




درباره Add Column نکات تکمیلی دیگری هم وجود دارد. گزینه Add Column را در درس Merge برای Cross-Join به کار بردیم. حالا می‌خواهیم ستون‌های سفارشی‌تر بسازیم. گاهی داده‌های یک ستون چند بخشی هستند؛ حروف دارند، اعداد دارند، کد دارند و ما نمی‌خواهیم از Split استفاده کنیم تا داده‌ها را جدا کنیم و در چند ستون بیاوریم. در ستون مورد نظر کلیک می‌کنیم، Add Column، سپس Column From Examples و From Selection را انتخاب می‌کنیم.



در اینجا انگار می‌خواهیم از یک AI استفاده کنیم. در واقع، ادعای مایکروسافت این است که در این بخش از هوش مصنوعی استفاده شده است. در ستون جدید، یک بخش از دیتای آن ستون را ما تایپ می‌کنیم و سیستم آن را تشخیص و اعمال فرمت یا الگو می‌کند و بقیه داده‌ها را مطابق نمونه پر می‌کند. برای بقیه سلول‌ها و سطرها، بقیه داده‌های درخواستی را سیستم پیشنهاد می‌دهد، اما رنگ آن‌ها کم‌رنگ است و در واقع یک پیشنهاد به ما ارائه می‌کند.



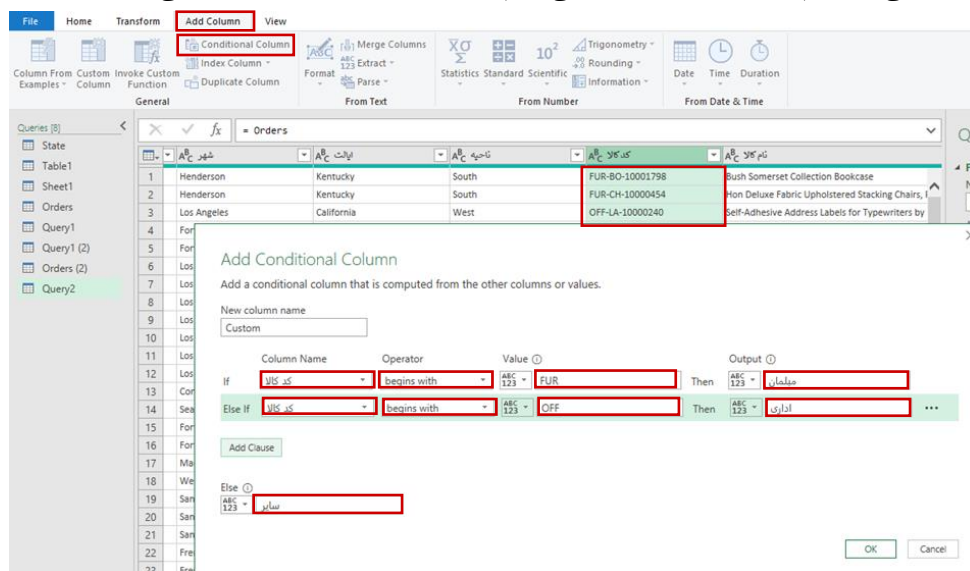
در واقع از نوشته‌ها می‌فهمیم که سیستم چه چیزی را مدنظر دارد و آن را در همه سطرهای دیگر نمایش می‌دهد. اگر درست بود، OK می‌زنیم و یک ستون جدید با داده‌های مدنظر ما تشکیل می‌شود.

اگر بخواهیم بر اساس تمام ستون‌های موجود یک چیزی را بنویسیم تا سیستم تشخیص دهد، گزینه Column From Examples → From All Columns را انتخاب می‌کنیم.

در بخش Custom Column نیز می‌توان فرمول‌نویسی انجام داد و ستون‌های جدید ساخت.

گاهی می‌خواهیم اطلاعات یک ستون و سلول را مشخص کنیم و مثلاً نوع کالا را تشخیص دهیم، بدون اینکه آن را به ستون‌های دیگر بشکنیم. در این صورت از گزینه Conditional Column استفاده می‌کنیم.

یک کادر جدید باز می‌شود. نام ستون جدید را تعیین می‌کنیم. یک شرط (If) برای آن نوشته می‌شود.



با این شرطی که می‌نویسیم تعیین می‌کنیم که اگر در ستون کد کالا یک سری اطلاعات عددی یا حروفی وجود داشت و مثلاً شامل F.U.R. بود، در ستون جدید مبلمان نوشته شود. همین‌طور می‌توانیم کدهای دیگری اضافه کنیم. در نهایت Else را مشخص می‌کنیم، مثلاً سایر.

به این صورت می‌توانیم داده‌ها را به صورت شرطی و با قوانین دلخواه، برای ساخت ستون جدید تعریف کنیم.

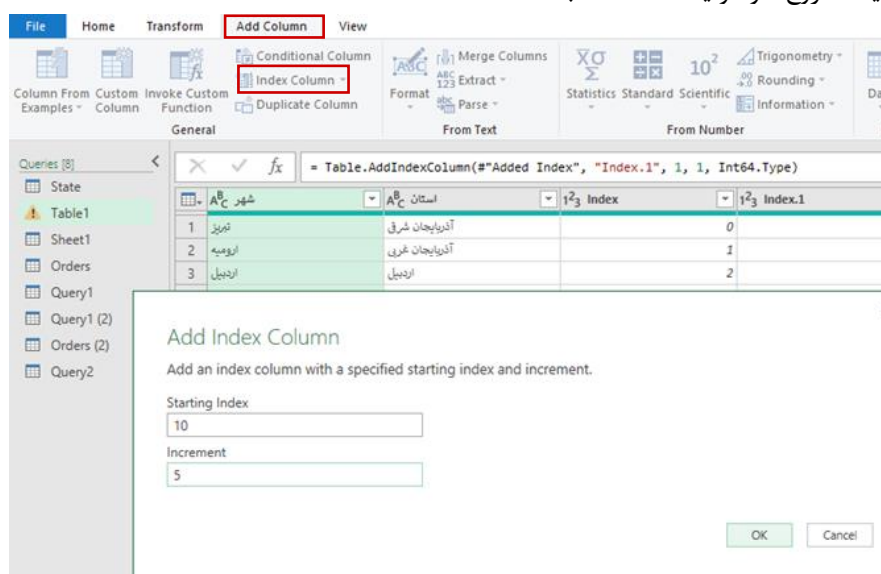
شماره کالا	نام کالا	نوع کالا
FUR-BO-10001798	Bush Somerset Collection Bookcase	مبل
FUR-CH-10000454	Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back	مبل
OFF-LA-10000240	Self-Adhesive Address Labels for Typewriters by Universal	اداری
FUR-TA-10000577	Bretford CR4500 Series Slim Rectangular Table	مبل
OFF-ST-10000760	Eldon Fold 'N Roll Cart System	اداری
FUR-FU-10001487	Eldon Expressions Wood and Plastic Desk Accessories, Cherry Wood	مبل
OFF-AR-10002833	Newell 322	اداری
TEC-PH-10002275	Mitel S320 IP Phone VoIP phone	سایر
OFF-BI-10003910	DXL Angle-View Binders with Locking Rings by Samsill	اداری
OFF-AP-10002892	Belkin F5C206VTEL 6 Outlet Surge	اداری
FUR-TA-10001539	Chromcraft Rectangular Conference Tables	مبل

اگر در Excel داده‌های جدید وارد شود، می‌توانیم باز هم یک شرط جدید تعریف کنیم و فایل را Refresh کنیم.

این کادر در Conditional Column همان زبان محاوره‌ای Power Query یا همان UI است، ولی در پشت صحنه با زبان M، کدنویسی نیز انجام می‌شود.

گاهی نیاز است به یک ستون شماره‌ی ترتیبی بدهیم، پس از گزینه Index Column. استفاده می‌کنیم این شماره

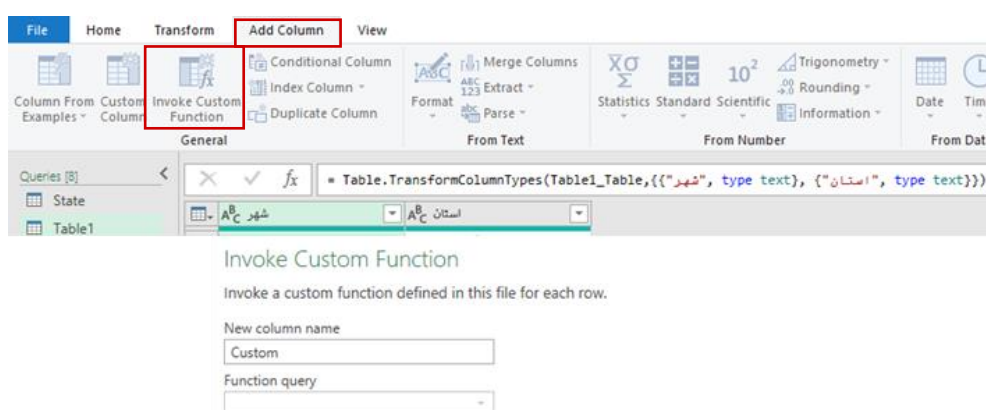
می‌تواند از صفر یا یک شروع شود و یا Custom باشد.



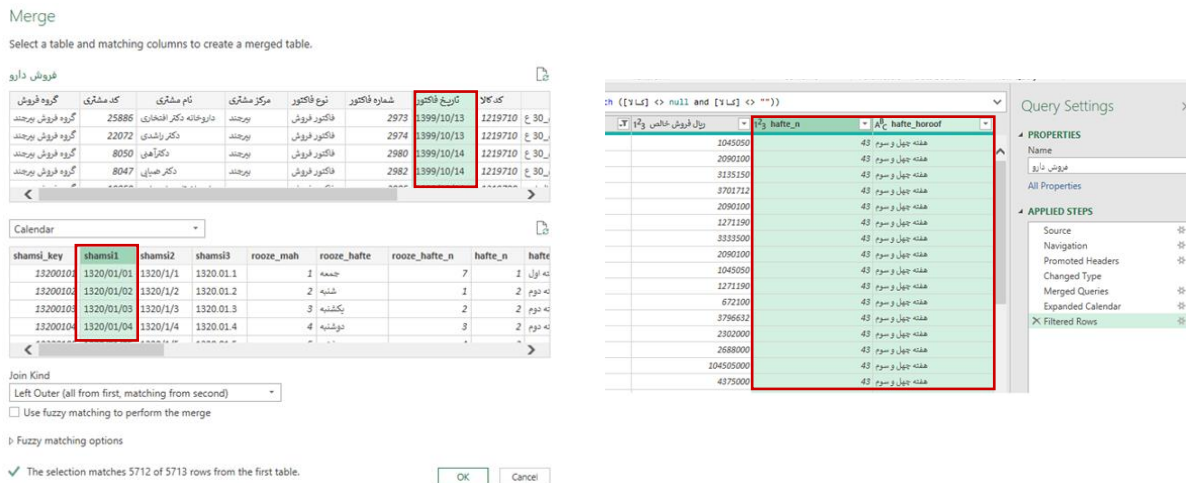
اگر در پاور کوئری Function نوشته باشیم یا از جایی وارد کرده باشیم، می‌توانیم با گزینه Invoke Custom

Function آن را فراخوانی کنیم و در فرمول بصورت دستی استفاده کنیم. از این گزینه برای اعمال Function آماده یا ایجاد

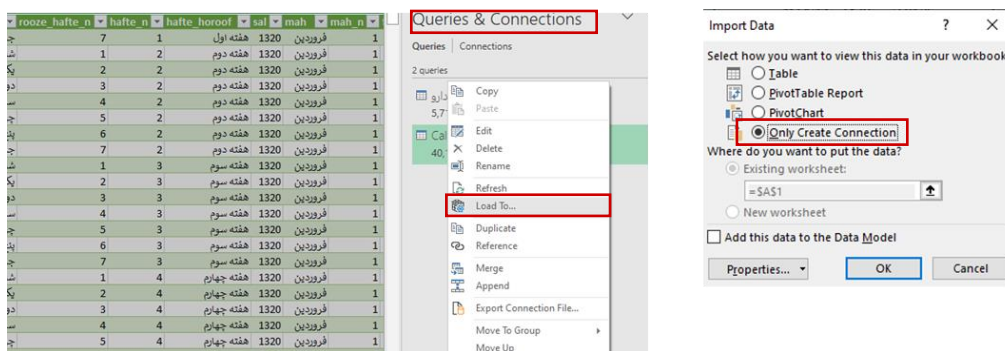
Function دلخواه در ستون‌ها استفاده می‌کنیم.



که ظاهر مشابهی دارد Merge می‌کنیم. سپس بر اساس نیاز، ستون‌هایی را که می‌خواهیم Expand می‌کنیم (برای مثال هفته‌ها) تا بتوانیم گزارش‌هایی، مثل جمع فروش بر اساس هفته در Pivot Table بگیریم.



بعد از انجام تغییرات در Power Query، Apply می‌کنیم و به Excel برمی‌گردیم. از بخش Queries روی Calendar راست‌کلیک کرده و گزینه‌ی Load To... را می‌زنیم و Only Create Connection را انتخاب می‌کنیم؛ یعنی فقط یک اتصال برقرار می‌شود و دیگر نیازی نیست تقویم ساخته‌شده در شیت اکسل نمایش داده شود.



نحوه ساخت (Dim Date) Date

ابتدا یک تاریخ میلادی می‌نویسیم، بعد با افزونه آن را به شمسی تبدیل می‌کنیم. سپس با توابع مختلف مثل IF، IFS، CHOOSE، VLOOKUP و غیره، ستون‌های دیگر را ایجاد می‌کنیم. این کار با استفاده از زبان M هم امکان‌پذیر است.

ما Constructor هایی داریم که با آن‌ها می‌توانیم Dim Date بسازیم. این روش در این جلسه توضیح داده می‌شود و روش دوم در جلسه دهم، با زبان M، آموزش داده خواهد شد.

تابع List.Dates هم تابعی است که از یک تاریخ تا یک تاریخ دیگر، مجموعه‌ای از تاریخ‌ها را تولید می‌کند.

برای شروع، یک New Query از نوع Blank Query ایجاد می‌کنیم، سپس Advanced Editor را باز می‌کنیم. حالا می‌خواهیم داخل آن Code بنویسیم.

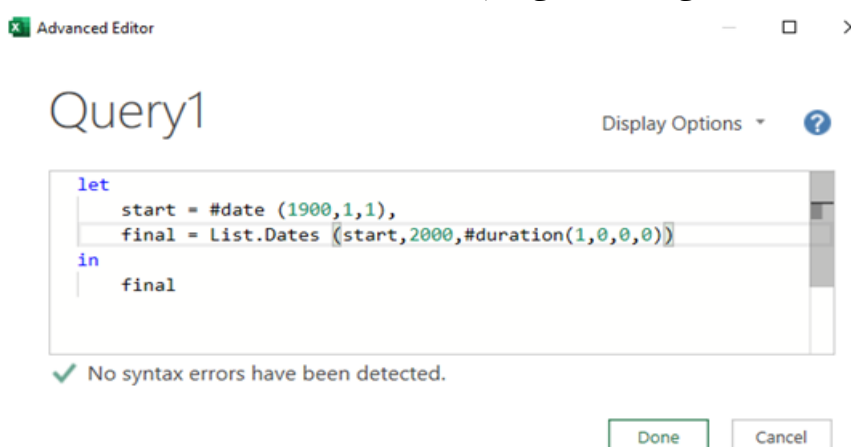
کد را با دستور let شروع می‌کنیم. بعد باید متغیری تعریف کنیم؛ مثلاً اینجا ما اسمش را Start می‌گذاریم، ولی هر نام دلخواه دیگری هم می‌توانیم انتخاب کنیم.

سپس مقدار اولیه را تعیین می‌کنیم — مثلاً مشخص می‌کنیم. دستورات را با کاما جدا می‌کنیم.

بعد متغیر Final را تعریف می‌کنیم. داخل آن از تابع List.Dates استفاده می‌کنیم. این تابع در واقع چهار Value دریافت می‌کند:

- تاریخ شروع را با تابع #date(Year, Month, Day)
- تعداد تاریخ‌هایی که باید بسازد (عدد مشخص می‌کنیم)
- #duration برای تعیین بازه (روز، ساعت، دقیقه، ثانیه)

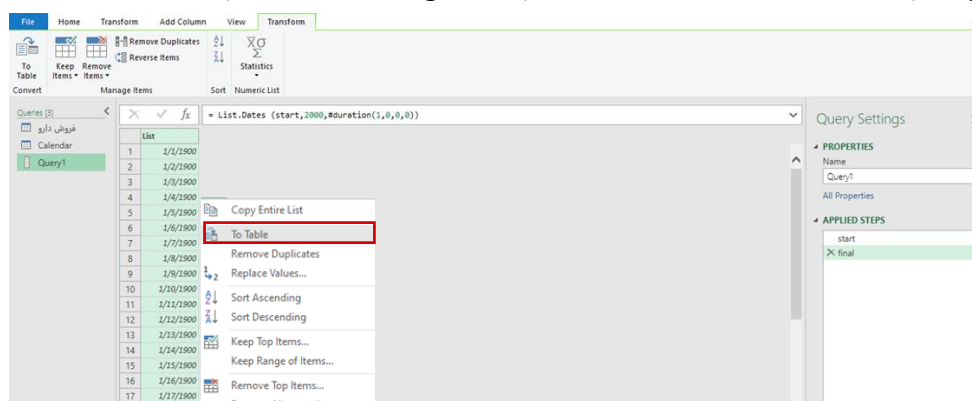
در نهایت با دستور in Final خروجی را مشخص می‌کنیم.



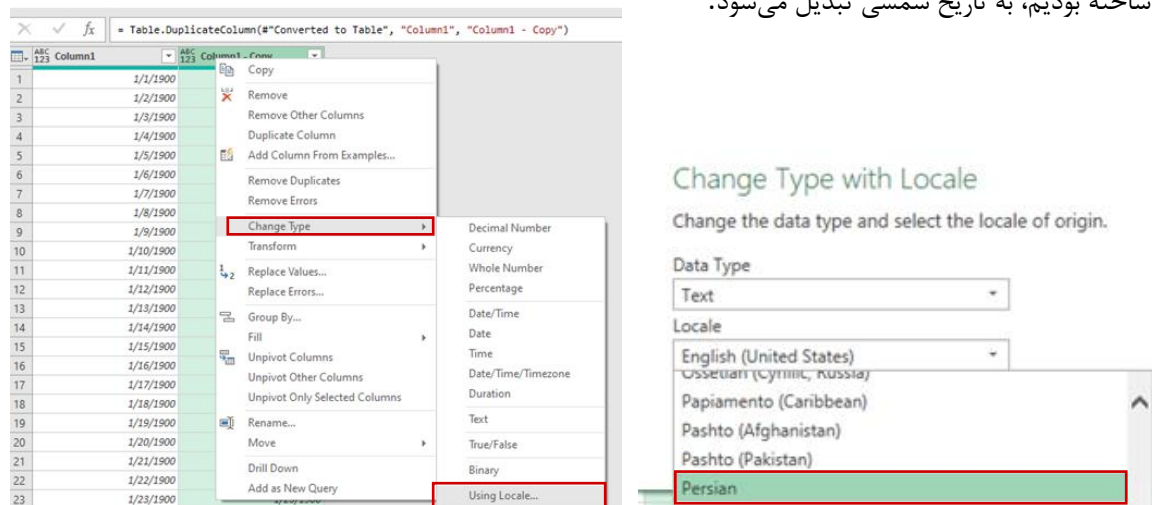
در Power Query، با اجرای این کد، یک List ساخته می‌شود که مثلاً شامل ۲۰۰۰ سطر است و از تاریخ شروع تا

پایان به صورت یکی‌یکی اضافه می‌شود. بعد با کلیک راست روی آن، گزینه‌ی Convert to Table را می‌زنیم تا تبدیل به

جدول شود. می‌توانیم ستون را Duplicate Column کنیم تا بعد نوع آن را مشخص کنیم.



در ادامه، با استفاده از گزینه‌ی Using Locale ، و انتخاب تنظیمات مربوط به Persian Calendar، تاریخ میلادی‌ای که با M ساخته بودیم، به تاریخ شمسی تبدیل می‌شود.



ما می‌توانیم هر سالی را که بخواهیم با کدنویسی در زبان M ایجاد کنیم، اما در Excel قبل از سال 1900 قابل شناسایی نیستند و نمی‌توانیم آن‌ها را نمایش دهیم.

بعد از ساخت این کدها، می‌توانیم آن‌ها را کپی کرده و در فایل‌ها یا پروژه‌های دیگر استفاده کنیم، یا حتی خود Query را مستقیماً کپی کنیم. این روش سریع‌تری برای ساخت Date Dimension است، چون فایل اکسلی که شامل Dim Date یا Calendar باشد ممکن است سنگین شود. در نتیجه، کدنویسی در زبان M یک مزیت محسوب می‌شود.

محاسبه فاصله بین تاریخ‌ها
اگر جدولی داشته باشیم و بخواهیم اختلاف بین دو تاریخ را در دو سطر محاسبه کنیم، معمولاً با افزودن یک ستون جدید (Add Column) و تعریف یک Custom Column، این کار را انجام می‌دهیم.

	A	B	C	D	E	F	G	H	I	J
جنسیت نوزاد	تاریخ تولد	تاریخ ازدواج والدین	تاریخ تولد پدر	تاریخ تولد مادر	تاریخ تنظیم سند					
دختر	1392/12/26	1386/06/11	1360/09/01	1370/08/05	1393/10/05					
پسر	1392/12/18	1389/04/12	1320/03/10	1353/03/03	1393/10/05					
پسر	1392/12/21	1382/09/13	1358/03/01	1364/01/19	1393/10/05					
دختر	1392/12/25	1376/01/25	1351/01/01	1358/06/10	1393/10/05					
دختر	1392/12/25	1391/06/08	1365/09/21	1365/01/13	1393/10/05					
پسر	1392/12/25	1389/12/22	1361/07/01	1366/09/15	1393/10/05					
پسر	1392/12/25	1390/11/20	1359/04/01	1372/01/25	1393/10/05					
دختر	1392/12/28	1388/05/14	1366/10/03	1355/06/03	1393/10/05					
دختر	1392/12/27	1388/11/03	1366/10/14	1372/01/30	1393/10/05					

اما اگر دو تاریخ را مستقیماً از هم کم کنیم، ممکن است در بعضی از سطرها Error بدهد. حتی اگر نوع داده را Date تنظیم کنیم، باز هم ممکن است درست تشخیص ندهد.

The Custom Column dialog shows the formula: `= [تاریخ تنظیم سند] - [تاریخ تولد]`. The available columns list includes: `چسبیت تولد`, `تاریخ تولد`, `تاریخ ازدواج والدین`, `تاریخ تولد پدر`, `تاریخ تولد مادر`, and `تاریخ تنظیم سند`. The table below shows a date subtraction resulting in an error.

ABC 123	تاریخ تولد	Custom	تاریخ ازدواج والدین
8	1392/12/28	Error	1388/05/14
9	1392/12/27	Error	1388/11/03
10	1392/12/26	Error	1390/03/26
11	1392/12/25	Error	1379/01/11
12	1392/12/26	Error	1379/05/06
13	1392/12/28	Error	1385/12/06
14	1393/01/03	Error	1388/08/25
15	1393/01/01	Error	1387/04/05
16	1392/12/27	Error	1388/07/30

Error message: `Expression.Error: We cannot apply operator - to types Text and Text. Details: Operator=-, Left=1393/01/05, Right=1392/12/25`

دلیل این خطا این است که Excel و Power Query تاریخ را بر اساس میلادی درک می کنند. مثلاً تاریخ «۱۳۹۳/۰۲/۳۱» وجود ندارند، چون در تقویم میلادی ماهی با ۳۱ روز و نیز در فوریه وجود ندارد. بنابراین مبنا همیشه تاریخ

The Transform Column Types dialog shows the formula: `= Table.TransformColumnTypes("#Reordered Columns",{{"تاریخ تولد", type date}})`. The table below shows a date being converted to a date type, resulting in an error.

ABC 123	تاریخ تولد	Custom	تاریخ ازدواج والدین
157	پسر	3/6/1393 Error	1387/01/16
158	دختر	3/6/1393 Error	1387/07/18
159	دختر	3/4/1393 Error	1389/08/06
160	دختر	Error	1380/04/01
161	پسر	3/5/1393 Error	1376/12/19
162	پسر	3/5/1393 Error	1376/12/19

Error message: `DataFormat.Error: We couldn't parse the input provided as a Date value. Details: 1393/02/31`

اگر از گزینه‌ی Using Locale هم استفاده کنیم و Persian را انتخاب کنیم، باز ممکن است مشکل باقی بماند، چون سیستم فقط نحوه نمایش (Format) را فارسی می کند، اما داده را همچنان به صورت میلادی در نظر می گیرد. در واقع، Locale فقط ظاهر را تغییر می دهد، نه محاسبات پشت آن را.

The Transform Column Types dialog shows the formula: `= Table.TransformColumnTypes("#Added Custom",{{"تاریخ تولد", type text}}, "fa-IR")`. The table below shows the same data with the locale set to Persian.

ABC 123	چسبیت تولد	تاریخ تولد	تاریخ ازدواج والدین	تاریخ تولد پدر
1	دختر	1392/12/26	1386/06/11	1360/09/01
2	پسر	1392/12/18	1389/04/12	1320/03/10
3	پسر	1392/12/21	1382/09/13	1358/03/01

اگر نوع داده به اشتباه Text باشد، Locale هم هیچ اثری ندارد. برای رفع این مشکل باید جدول Dim Date را وارد کنیم و با جدول اصلی Merge کنیم.

باید ستون تاریخ جدول اصلی را با ستون تاریخ Dim Date که ظاهر مشابهی دارد Merge می‌کنیم. از جدول Dim Date ستون معادل میلادی را انتخاب و Expand می‌کنیم. این کار را برای هر دو ستون تاریخ انجام می‌دهیم (مثلاً تاریخ شروع و تاریخ پایان) سپس در Add Column → Custom Column، اختلاف دو تاریخ میلادی را از هم کم می‌کنیم تا فاصله بین آن‌ها به دست آید.

Merge

Select a table and matching columns to create a merged table.

والات

چشمیت تولد	تاریخ تولد	تاریخ ازدواج والدین	تاریخ تولد پدر	تاریخ تولد مادر	تاریخ تنظیم سند
دختر	1392/12/26	1386/06/11	1360/09/01	1370/08/05	1393/01/05
پسر	1392/12/18	1389/04/12	1320/03/10	1353/03/03	1393/01/05
پسر	1392/12/21	1382/09/13	1358/03/01	1364/01/19	1393/01/05
دختر	1392/12/25	1376/10/25	1351/01/01	1358/06/10	1393/01/05
دختر	1392/12/25	1391/06/08	1365/09/21	1365/01/13	1393/01/05

Calendar

Custom Column

Add a column that is computed from the other columns.

New column name

Custom

Custom column formula

= [تاریخ تنظیم سند میلادی] - [تاریخ تولد میلادی]

Available columns

چشمیت تولد
تاریخ تولد
تاریخ ازدواج والدین
تاریخ تولد پدر
تاریخ تولد مادر
تاریخ تنظیم سند
تاریخ تولد میلادی
تاریخ تنظیم سند میلادی

<< Insert

Learn about Power Query formulas

No syntax errors have been detected

OK Cancel

به این ترتیب، اختلاف تاریخ‌ها به درستی محاسبه می‌شود و از خطاهای مربوط به تقویم شمسی و Locale جلوگیری می‌شود.

جلسه نهم

مثال های کاربردی

در این جلسه ۸ مثال کاربردی را در پاور کوئری بررسی می کنیم.

مسئله اول -

جدولی که داریم باید داده ها در قالب استاندارد باشند تا دسترسی به آن ها راحت تر شود. در ستون اول، میانگین تأخیر را نمی خواهیم، زیرا این ستون مربوط به «ناحیه» است. میانگین تأخیر را می توانیم بعداً در یک ستون جدید از روی «مدت» محاسبه کنیم، پس باید حذف شود.

ناحیه	نام و تاریخ	تعداد	مدت (روز)
شمال	خرام - ۱۴۰۲/۰۲/۰۹	۴	۸
شمال	فرهادی - ۱۴۰۲/۰۲/۰۹	۳	۸
شمال	مستحق - ۱۴۰۲/۰۱/۳۰	۱	۹
شمال	اعتقادی - ۱۴۰۲/۰۲/۰۱	۲	۷
شمال	پورگری - ۱۴۰۲/۰۱/۳۱	۳	۵
میانگین تأخیر			۷.۴
مرکز	حسینی - ۱۴۰۲/۰۲/۰۵	۱	۹
مرکز	مهری - ۱۴۰۲/۰۲/۰۶	۲	۸
مرکز	فرهاد - ۱۴۰۲/۰۲/۰۵	۳	۴
مرکز	حمیدی رفیع - ۱۴۰۲/۰۲/۰۴	۳	۱۴
مرکز	رجعی - ۱۴۰۲/۰۲/۰۳	۱	۶
میانگین تأخیر			۸.۲

زیرا در سطر «میانگین تأخیر»، بقیه ستون ها هم مقدار null دارند، پس باید حذف شوند. برای این کار، یا باید از ستون اول، سطرهایی را که مربوط به میانگین تأخیر هستند Filter کنیم، (یا می توانیم با نوشتن یک کد با if و and این کار را انجام دهیم). اگر گزینه ی Remove Empty را بزنیم، مقادیر null حذف می شوند اما در این حالت نام ناحیه ها را هم از دست می دهیم، چون زیر هر ناحیه هم سطرهایی وجود داشته که null بوده اند. بنابراین باید یک Step قبل از آن انجام دهیم. در این مرحله گفتیم که گاهی ترتیب Step ها مهم است، پس باید برای سلول پایینی هر ناحیه از گزینه ی Fill Down استفاده کنیم تا ابتدا null ها با نام ناحیه ی خودشان پر شوند، و بعد از آن گزینه ی Remove Empty را اجرا کنیم.

ناحیه	نام و تاریخ	تعداد	مدت (روز)
شمال	خرام - ۱۴۰۲/۰۲/۰۹	۴	۸
شمال	فرهادی - ۱۴۰۲/۰۲/۰۹	۳	۸
شمال	مستحق - ۱۴۰۲/۰۱/۳۰	۱	۹
شمال	اعتقادی - ۱۴۰۲/۰۲/۰۱	۲	۷
شمال	پورگری - ۱۴۰۲/۰۱/۳۱	۳	۵
میانگین تأخیر			۷.۴
مرکز	حسینی - ۱۴۰۲/۰۲/۰۵	۱	۹
مرکز	مهری - ۱۴۰۲/۰۲/۰۶	۲	۸
مرکز	فرهاد - ۱۴۰۲/۰۲/۰۵	۳	۴
مرکز	حمیدی رفیع - ۱۴۰۲/۰۲/۰۴	۳	۱۴
مرکز	رجعی - ۱۴۰۲/۰۲/۰۳	۱	۶
میانگین تأخیر			۸.۲

برای جداسازی ستون دوم، می‌توانیم از گزینه‌ی Split Column by Delimiter استفاده کنیم تا نام‌ها را از تاریخ‌ها جدا کنیم.

ناحیه	نام	تاریخ	تعداد	مدت (روز)	
شمال	جرفام	1403/02/09	4		8
شمال	فرهادی	1403/02/09	3		8
شمال	ممشان	1403/01/30	1		9
شمال	اعتقانی	1403/02/01	2		7
شمال	پورگری	1403/01/31	3		5
مرکز	حسینی	1403/03/05	1		9
مرکز	مهری	1403/03/06	2		8
مرکز	فرجاد	1403/03/05	2		4
مرکز	حمیدنی رفیع	1403/03/04	3		14
مرکز	رنجی	1403/03/03	1		6
جنوب	مزدانی	1403/01/25	2		11
جنوب	مزارجی	1403/01/26	2		15
جنوب	رضاایی	1403/01/20	3		2
جنوب	بیمانی	1403/01/29	3		15
جنوب	حرفی	1403/01/28	3		2

در پایان، فایل آماده می‌شود و اگر در Excel داده‌ها به آن اضافه شوند و فایل به‌روزرسانی شود، همه‌ی داده‌ها در همین قالب و با تمام این تنظیمات جدید اضافه خواهند شد.

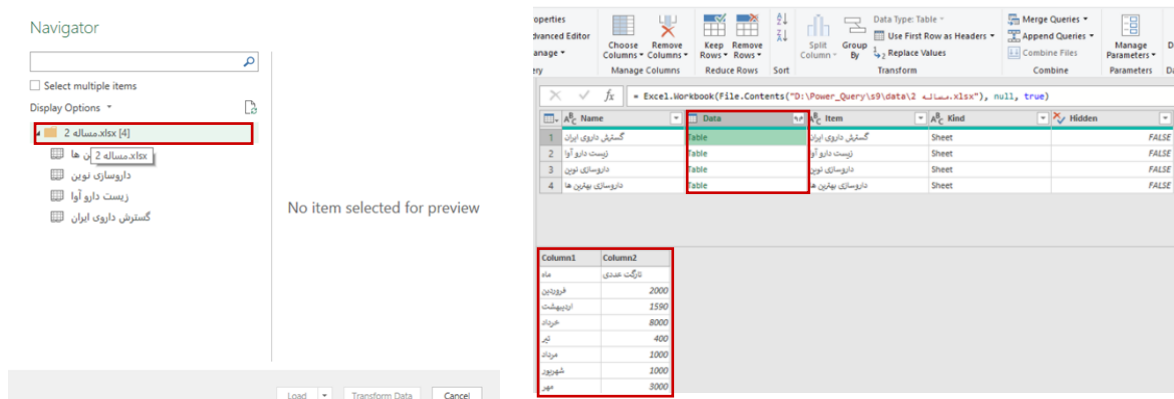
مساله ۲-

می‌خواهیم همه‌ی Sheet ها یکی شوند و داده‌ها در یک جدول قرار بگیرند.

ماه	تاریخ عددی
فروردین	۳۰۰۰
اردیبهشت	۱۵۹۰
خرداد	۸۰۰۰
تیر	۴۰۰
مرداد	۱۰۰۰
شهریور	۱۰۰۰
مهر	۳۰۰۰
آبان	۶۰۰۰
آذر	۴۰۰۰
دی	۳۰۰۰
بهمن	۲۰۰۰

در این حالت می‌توان از گزینه‌ی Import Folder هم استفاده کرد، اما کاری که می‌خواهیم انجام دهیم با روش دیگری در Power Query قابل انجام است. در اینجا باید روی گزینه فولدر کلیک کنیم تا همه فایل‌های اکسلی که درون خود دارد را در Power Query بیاورد.

در Power Query وقتی فایل Excel را Import می‌کنیم، لیستی از تمام Sheet های آن فایل نمایش داده می‌شود.

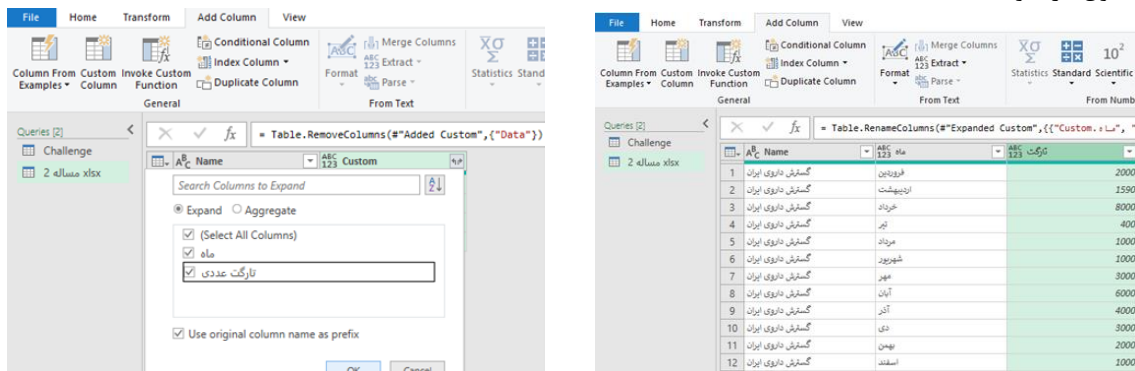


جدولی که در این مرحله نشان داده می‌شود، شامل ستون‌هایی مانند Name، Data و ... است. در پایین صفحه در ستون Kind مشخص می‌شود که هر مورد مربوط به Sheet یا Table یا Text است. در اینجا می‌توانیم از گزینه‌ی Text Filter استفاده کنیم و مقدار آن را برابر با Sheet قرار دهیم تا فقط شیت‌ها نمایش داده شوند. به این ترتیب مطمئن می‌شویم فقط داده‌های موردنظرمان در ادامه پردازش می‌شوند. سپس دو ستون غیرضروری را Remove می‌کنیم. بین ستون‌های Name و Item یکی را نگه می‌داریم و دیگری را حذف می‌کنیم (معمولاً ستون Name را نگه می‌داریم) در ستون Data، داده‌ها به‌صورت جدول کوچک (Table) در پایین صفحه نمایش داده می‌شوند. گاهی در این جدول‌ها، سرستون‌ها درست شناسایی نمی‌شوند.

برای رفع این مشکل، یک Custom Column ایجاد می‌کنیم و در آن از تابع زیر استفاده می‌کنیم:

`Table.PromoteHeaders([Data])`

با این کار، سرستون‌ها به‌درستی شناسایی می‌شوند و همه‌ی Table ها ساختار یکسانی پیدا می‌کنند. بعد از آن، ستون Data را دیگر نیاز نداریم، پس آن را حذف می‌کنیم. از ستون Custom گزینه‌ی Expand را انتخاب می‌کنیم تا محتویات هر جدول باز شود.



در نتیجه، یک جدول نهایی خواهیم داشت که شامل ستون‌های مثلاً نام، ماه، تارگت و... است. به این ترتیب همه‌ی شیت‌ها در یک جدول واحد و مرتب آماده می‌شوند.

نکته مهم این است که اگر در فایل Excel شیت جدیدی اضافه شود، با زدن Refresh در Power Query، داده‌های شیت جدید هم به‌صورت خودکار در جدول نهایی نمایش داده می‌شوند.

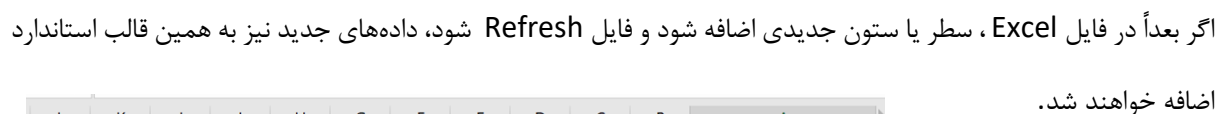
مسئله ۳ – اصلاح ساختار جدول (Unpivot Columns) در Power Query

در این مسئله، جدولی داریم که جای سطرها و ستون‌ها درست نیست. سرستون‌ها باید نام فیلدها (Field Name) باشند، نه مقدار فیلدها. (Field Value)

	فروردین	اردیبهشت	خرداد	تیر	مرداد	شهریور	مهر	آبان	آذر	دی	بهمن	اسفند
گسترش داروی ایران	2000	1590	8000	400	1590	3000	1200	1500	23000	900	2500	2500
زیست دارو آوا	2000	1590	8000	400	1590	3000	1200	1500	23000	900	2500	2500
داروسازی نوین	2000	1590	8000	400	1590	3000	1200	1500	23000	900	2500	2500

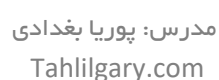
برای مثال: اگر هر ستون نام یک ماه باشد و سطرها نام شرکت‌ها باشند، در واقع ستون‌های ماه‌ها «مقادیر» هستند و باید به فیلد تبدیل شوند تا جدول ساختار درستی بگیرد. به بیان دیگر، نام ماه‌ها باید در سطرها قرار بگیرند، نه در ستون‌ها. روش انجام کار اینطور است که اگر فقط یک ستون درست داریم (مثلاً نام شرکت‌ها)، باید روی همان ستون Right Click کنیم و گزینه‌ی Unpivot Other Columns را بزنیم. با این کار، تمام ستون‌های دیگر که مربوط به ماه‌ها بودند، به سطرها تبدیل می‌شوند و داده‌ها به‌صورت استاندارد در می‌آیند.

در نتیجه، ستون‌هایی مانند Attribute (مثلاً نام ماه) و Value (مثلاً مقدار فروش) ساخته می‌شوند.



شماره	نام شرکت	سال	رتبه
1	گسترش داروی ایران	فروردین 2000	2
2	گسترش داروی ایران	اردیبهشت 1590	3
3	گسترش داروی ایران	خرداد 8000	4
4	گسترش داروی ایران	تیر 400	5
5	گسترش داروی ایران	مرداد 1000	6
6	گسترش داروی ایران	شهریور 1000	7
7	گسترش داروی ایران	مهر 3000	8
8	گسترش داروی ایران	آبان 6000	9
9	گسترش داروی ایران	آذر 4000	10
10	گسترش داروی ایران	دی 3000	11
11	گسترش داروی ایران	بهمن 2000	12

برای رفع این مشکل، یا باید در Step مربوط به Change Type در بخش Fomula bar کد مربوط به Change Type اسفند را حذف کنیم و یا آنکه بصورت کلی آن Step را حذف کنیم. و نهایتاً نوع داده‌ی ستون نهایی (مثلاً ستون Target) را خودمان تنظیم کنیم، مثلاً روی Whole Number قرار دهیم.



گاهی در داده‌های جدید ممکن است یکی از ستون‌هایی که قبلاً Unpivot کرده بودیم، دیگر وجود نداشته باشد. در این حالت، اگر بخواهیم کد M را هر بار تغییر دهیم، کار زمان‌بر و خسته‌کننده می‌شود. در چنین شرایطی، زبان M به کمک ما می‌آید. می‌توانیم در کد M بنویسیم که از بین مجموعه ستون‌هایی که مشخص کرده‌ایم، فقط آن‌هایی را Unpivot کند که واقعاً در داده وجود دارند.

به این ترتیب، هنگام به‌روزرسانی (Refresh)، Power Query خودش با توجه به تغییرات فایل، به‌صورت پویا عمل می‌کند و نیازی به ویرایش دستی کد نخواهد بود.

مسئله ۴ - رفع ادغام سلول‌ها و تبدیل داده‌های سطری به ستونی در Power Query

در بعضی داده‌ها، سلول‌ها Merge شده‌اند. در این حالت، فقط سلول اول حاوی متن است و بقیه سلول‌های ادغام‌شده مقدار null نمایش می‌دهند.

تاریخ عددی	ماه
۲۰۰۰	فروردین
۱۳۹۰	اردیبهشت
۸۰۰۰	خرداد
۳۰۰۰	تیر
۱۰۰۰	مرداد
۱۰۰۰	شهریور
۳۰۰۰	گسترش داروی ایران
۶۰۰۰	آبان
۴۰۰۰	آذر
۳۰۰۰	دی
۴۰۰۰	بهمن
۱۰۰۰	اسفند
۳۳۰۰	فروردین
۳۰۰۰	اردیبهشت
۱۲۰۰	خرداد
۱۵۰۰	تیر
۱۵۰۰	مرداد
۵۶۰۰	شهریور
۱۲۰۰	آبان

برای رفع این مشکل، از تب Transform گزینه Fill Down را می‌زنیم. با این کار، سلول‌های null با مقدار سلول بالایی خود پر می‌شوند و جدول ساختار صحیحی پیدا می‌کند.

تاریخ عددی	ماه
۲۰۰۰	فروردین
۱۳۹۰	اردیبهشت
۸۰۰۰	خرداد
۴۰۰	تیر
۱۰۰۰	مرداد
۱۰۰۰	شهریور
۳۰۰۰	گسترش داروی ایران
۶۰۰۰	آبان
۴۰۰۰	آذر
۳۰۰۰	دی
۲۰۰۰	بهمن
۱۰۰۰	اسفند
۲۳۰۰۰	فروردین
۳۰۰۰	اردیبهشت
۱۲۰۰	خرداد
۱۲۰۰	تیر

به خاطر داریم که در Power Query دو نوع Fill داریم Fill Down و Fill Up ، اما گزینه‌های Fill Left یا Fill Right وجود ندارند.

مسئله ۵

در این جدول، یک ستون داریم که داده‌های آن یکی در میان تغییر می‌کنند؛ یعنی یک سطر نام ماه است و سطر بعدی مقدار مربوط به آن ماه. به عبارتی، در هر دو سطر، نوع داده عوض می‌شود.

عنوان	مقدار
ماه	فروردین
درخواست	۲۱۵
ماه	اردیبهشت
درخواست	۲۰۱
ماه	خرداد
درخواست	۲۲۸
ماه	تیر
درخواست	۲۸۸
ماه	مرداد
درخواست	۲۱۸

عنوان	مقدار
ماه	فروردین
درخواست	۲۱۵
ماه	اردیبهشت
درخواست	۲۰۱
ماه	خرداد
درخواست	۲۲۸
ماه	تیر
درخواست	۲۸۸
ماه	مرداد
درخواست	۲۱۸
ماه	مهر
درخواست	۲۸۱

برای حل این مشکل چند مرحله انجام می‌دهیم:

مرحله ۱: ایجاد ستون Index

از تب From 0 → Index Column → Add Column یک ستون ایندکس از عدد ۰ ایجاد می‌کنیم.

مرحله ۲: افزودن ستون تقسیم صحیح

روی ستون Index کلیک می‌کنیم، از تب Divide (Integer Division) → Standard → Add Column را انتخاب

می‌کنیم. در کادر باز شده عدد 2 را وارد می‌کنیم.

در نتیجه، اعداد ایندکس به این صورت تقسیم می‌شوند:

- ایندکس ۰ و ۱ مقدار ۰ می‌گیرند
- ایندکس ۲ و ۳ مقدار ۱ می‌گیرند

- ایندکس ۴ و ۵ مقدار ۲ می‌گیرند و به همین ترتیب تا آخر.

به این ترتیب، هر دو سطر زیر یک گروه مشترک قرار می‌گیرند.

مرحله ۳: افزودن ستون باقی‌مانده (Modulo)

دوباره روی ستون Index کلیک می‌کنیم، از مسیر Add Column → Standard → Modulo را انتخاب کرده و عدد 2 را وارد می‌کنیم. اکنون ستونی داریم که مقادیرش یکی در میان ۰ و ۱ هستند. سطرهایی با مقدار ۰ نام ماه‌ها را نشان می‌دهند و سطرهایی با مقدار ۱ مقدار مربوط به همان ماه را دارند.

The screenshot shows the Power Query Editor interface. The 'Add Column' menu is open, and 'Modulo' is selected. Below the menu, there are two dialog boxes. The first is 'Integer-Divide' with a 'Value' field containing '2'. The second is 'Modulo' with a 'Value' field containing '2'. The background shows a table with columns: عنوان, Index, Integer-Division, and Modulo. The 'Modulo' column has values 0 and 1 alternating.

مرحله ۴: حذف ستون Index

چون دیگر به آن نیاز نداریم، ستون Index را حذف می‌کنیم

مرحله ۵: انجام Pivot

اکنون باید ستون Modulo را Pivot کنیم. در این: Pivot

- سطرهایی با مقدار ۰، نام ماه‌ها هستند.

- سطرهایی با مقدار ۱، مقادیر همان ماه‌ها هستند.

در پنجره باز شده، در بخش Values Column، ستونی را انتخاب می‌کنیم که شامل داده‌های اصلی است (یعنی همان ستونی که ماه و مقدار در آن یکی در میان آمده بود).

در بخش Advanced Options گزینه Don't Aggregate را فعال می‌کنیم، چون نمی‌خواهیم داده‌ها تجمیع شوند.

Use the names in column "Modulo" to create new columns.

عنوان = مقدار

Aggregate Value Function

Don't Aggregate

[Learn more about Pivot Column](#)

OK

Cancel

Item

Add Column

View

anspose

Reverse Rows

Count Rows

Data Type: Any

Detect Data Type

Rename

Replace Values

Fill

Pivot Column

Any Column

Unpivot Columns

Move

Convert to List

Split Column

Format

Text Column

ASC

46%

123

Extract

Parse

Σ

Statistics

Standard

Scientific

Number Column

10^2

Rounding

Information

Trigonometry

Expand

Aggregate

Extract Values

Structured Column

Date

Time

Duration

Date & Time Column

×

✓

fx

Table.Pivot(Table.TransformColumnTypes(#"Removed Columns", {{"Modulo", type text}}, "en-US"), List.Distinct

Integer-Division

0

1

1	0	فروردین	215
2	1	اردیبهشت	201
3	2	خرداد	248
4	3	تیر	288
5	4	مرداد	218
6	5	شهریور	281
7	6	مهر	234
8	7	آبان	241
9	8	آذر	265
10	9	دی	223
11	10	بهمن	273
12	11	اسفند	null

Query Settings

PROPERTIES

Name

Sheet1

All Properties

APPLIED STEPS

Source

Navigation

Promoted Headers

Changed Type

Removed Columns

Added Index

Inserted Integer-Division

Inserted Modulo

Removed Column1

Removed Column2

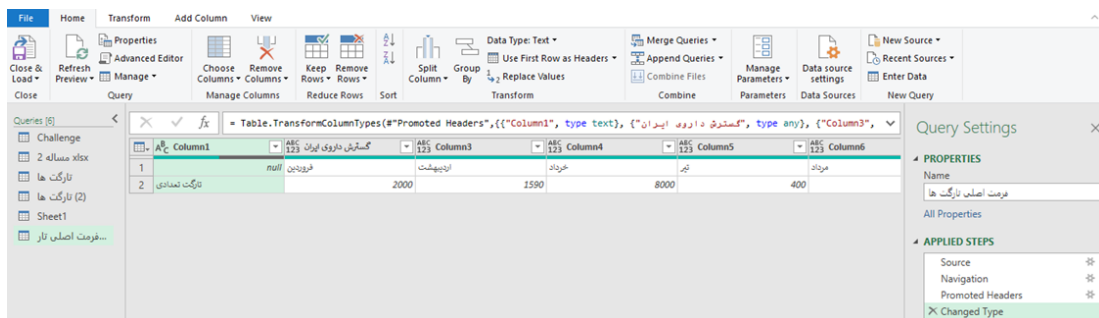
در نهایت، دو ستون جدید به دست می‌آید: یکی شامل نام ماه‌ها و دیگری شامل مقدار هر ماه در سطر مقابل آن. در پایان، فقط کافی است سرستون‌ها را **Rename** کنیم. به این ترتیب، داده‌هایی که در ابتدا ساختار نامنظمی داشتند، به‌صورت استاندارد و قابل تحلیل در **Power Query** تنظیم می‌شوند.

در واقع این یک تکنیک است برای داده‌های پیکه در میان متفاوت که میخواهند دو ستون بشوند.

مسئله ۶ - تبدیل جدول شامل نام شرکت‌ها و ماه‌ها به ساختار ستونی استاندارد در Power Query

در این مسئله، داده‌ها به صورت زیر هستند:

- در سطر اول نام یک شرکت در ستون اول آمده است.
- در ستون‌های بعدی، مقادیر مربوط به دوازده ماه وجود دارند.
- سپس در سطر پایین‌تر، نام شرکت بعدی تکرار شده و دوباره داده‌های ماه‌های آن قرار دارند.



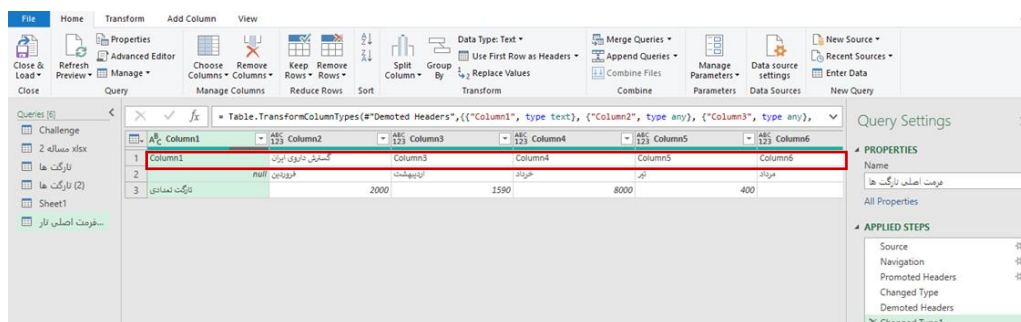
هدف ما این است که این جدول را به سه ستون استاندارد تبدیل کنیم: نام شرکت، نام ماه، مقدار.

مرحله ۱: حذف عنوان‌های اولیه از تیترها

ابتدا باید جدول را از حالت فعلی که سطر اول تیتراست خارج کنیم. برای این کار از تب Home

گزینه‌ی Use Headers as First Row را می‌زنیم. به این ترتیب، تیترها به یک سطر پایین‌تر می‌آیند و داده‌ها در موقعیت

درست قرار می‌گیرند.

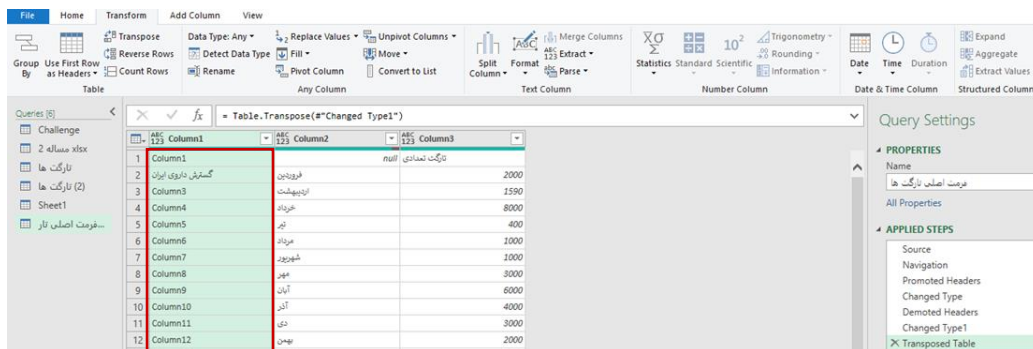


مرحله ۲: جابه‌جایی سطر و ستون (Transpose)

اکنون باید داده‌ها را بچرخانیم تا ساختارشان درست شود. چون ما Fill Right نداریم، برای این کار از گزینه‌ی

Transform → Transpose استفاده می‌کنیم. دستور Transpose باعث می‌شود سطرها و ستون‌ها جایشان را با یکدیگر

عوض کنند.



مرحله ۳: پر کردن مقادیر خالی شرکتها

پس از Transpose، در سطر اول نام شرکتها وجود دارد، اما در سطرهای بعدی شماره ستون آمده است، پس null نیستند، برای رفع این مشکل باید با Conditional Column ستون شرطی بسازیم. برای مثال: اگر ستون اول شامل کلمه‌ای خاص بود (مثلاً "Column" یا عنوان غیر واقعی)، می‌توانیم تعیین کنیم خروجی آن سطر null باشد؛ در غیر این صورت مقدار واقعی آن نمایش داده شود

Add Conditional Column
Add a conditional column that is computed from the other columns or values.

New column name
Custom

Column Name Operator Value Output
If Column1 contains ABC 123 Column Then ABC 123 null

Add Clause

Else
Column1
Enter a value
Select a column
Parameter

ABC 123	Column1	ABC 123	Column2	ABC 123	Column3	ABC 123	Custom
1	Column1						null
2	گسترش داروی ایران	فروزمین		2000			گسترش داروی ایران
3	گسترش داروی ایران	ازدیهشت		1590			null
4	Column4	خرداد		8000			null
5	Column5	نیر		400			null
6	Column6	مرداد		1000			null
7	Column7	شهرزور		1000			null
8	Column8	مهر		3000			null
9	Column9	آبان		6000			null
10	Column10	آذر		4000			null
11	Column11	دی		3000			null
12	Column12	بهمن		2000			null
13	Column13	اسفند		1000			null
14	زیست دارو آوا	فروزمین		23000			زیست دارو آوا
15	Column15	ازدیهشت		3000			null
16	Column16	خرداد		1200			null

سپس روی این ستون جدید، دوباره Fill Down می‌زنیم تا مقادیر null پر شوند. به این ترتیب، همه سطرها نام شرکت مربوط به خود را خواهند داشت. برخی سطرها در ستون‌های دیگر هم ممکن است مقدار Null داشته باشند که با Remove Empty می‌توان آنها را هم رفع کرد.

در نهایت، ستون‌ها را به نام‌های مناسب تغییر می‌دهیم، به این ترتیب، جدول ما استاندارد و آماده‌ی تحلیل می‌شود— به‌طوری‌که هر سطر شامل یک شرکت، یک ماه، و مقدار مربوط به آن است.

fx = Table.RenameColumns(#"Filtered Rows",{"Custom","شرکت"},{"Column2","ماه"},{"Column3","تراکت"})

شرکت	ماه	تراکت
گسترش داروی ایران	فروزمین	2000
گسترش داروی ایران	ازدیهشت	1590
گسترش داروی ایران	خرداد	8000
گسترش داروی ایران	نیر	400
گسترش داروی ایران	مرداد	1000
گسترش داروی ایران	شهرزور	1000
گسترش داروی ایران	مهر	3000
گسترش داروی ایران	آبان	6000
گسترش داروی ایران	آذر	4000
گسترش داروی ایران	دی	3000
گسترش داروی ایران	بهمن	2000
گسترش داروی ایران	اسفند	1000
زیست دارو آوا	فروزمین	23000
زیست دارو آوا	ازدیهشت	3000
زیست دارو آوا	خرداد	1200
زیست دارو آوا	نیر	1500
زیست دارو آوا	مرداد	15000

Query Settings

PROPERTIES
Name
فرمت اصلی تراکت ها
All Properties

APPLIED STEPS
Source
Navigation
Promoted Headers
Changed Type
Demoted Headers
Changed Type1
Transposed Table
Added Conditional Column
Removed Columns
Reordered Columns
Filled Down
Filtered Rows

مسئله ۷ - جدول با دو هدر (دو ردیف عنوان)

در این جدول، داده‌ها به صورت دو ردیف عنوان (Header) و چند ردیف داده نمایش داده می‌شوند. یعنی به جای یک ردیف عنوان در بالا، دو ردیف داریم که ترکیب آن‌ها معنی هر ستون را مشخص می‌کند.

The screenshot shows an Excel spreadsheet with a table containing sales data. The table has two header rows. The first header row contains: نام شرکت (Company Name), بهار (Spring), تابستان (Summer), پائیز (Autumn), and زمستان (Winter). The second header row contains: فروش بهار (Spring Sales), فروش تابستان (Summer Sales), فروش پائیز (Autumn Sales), and فروش زمستان (Winter Sales). The data rows follow, with columns for company name and sales figures for each season.

Below the table, the Power Query formula bar shows the formula: `= Table.TransformColumnTypes(#Promoted Headers,{"نام شرکت", type text}, {"بهار", type any}, {"Column3", type any}, {"Column4", type any}, {"Column5", type any}, {"Column6", type any})`. The Query Settings pane on the right shows the applied steps: Source, Navigation, Promoted Headers, and Changed Type.

در ابتدا باید داده‌های ستون اول را از حالت Header خارج کنیم تا بتوانیم روی آن‌ها عملیات انجام دهیم. از تب: Transform → Use Headers → Use Headers as First Row استفاده می‌کنیم. سپس از تب Transpose استفاده می‌کنیم تا سطرها و ستون‌ها جایشان را عوض کنند. این کار باعث می‌شود داده‌ها به فرم عمودی دربیایند و کار روی آن‌ها راحت‌تر شود.

The screenshot shows the Excel spreadsheet after the Transpose operation. The table is now oriented vertically. The first column contains the company names, and the subsequent columns contain the sales figures for each season. The Power Query formula bar shows the formula: `= Table.Transpose(#"Changed Type1")`. The Query Settings pane on the right shows the applied steps: Source, Navigation, Promoted Headers, Changed Type, Demoted Headers, and Transposed Table.

در این مرحله، هنوز ستون‌های مربوط به نام شرکت‌ها به صورت افقی هستند. برای اینکه آن‌ها را به سطر تبدیل کنیم، از تب: Transform → Use First Row as Headers استفاده می‌کنیم تا ردیف اول عنوان شود، سپس ستون شرکت‌ها را انتخاب کرده و از تب: Transform → Unpivot Columns اجرا می‌کنیم.

form Add Column View

Conditional Column Index Column Duplicate Column

Format Parse From Text

Statistics Standard Scientific From Number

Trigonometry Rounding Information From Date

fx = Table.RenameColumns(#"Unpivoted Columns",{{"Column2", "ماه"}, {"Attribute", "شرکت"}}

	فصل	ماه	شرکت	تارگت
1	بهار	فروردین	گسترش داروی ایران	2000
2	بهار	فروردین	زیست دارو آوا	23000
3	بهار	فروردین	داروسازی نوین	900
4	Column3	اردیبهشت	گسترش داروی ایران	1590
5	Column3	اردیبهشت	زیست دارو آوا	3000
6	Column3	اردیبهشت	داروسازی نوین	2500
7	Column4	خرداد	گسترش داروی ایران	8000
8	Column4	خرداد	زیست دارو آوا	1200
9	Column4	خرداد	داروسازی نوین	2500
10	تابستان	تیر	گسترش داروی ایران	400
11	تابستان	تیر	زیست دارو آوا	1500
12	تابستان	تیر	داروسازی نوین	2000

مانند مسئله قبل (شماره ۶)، حالا باید با یک Conditional Column داده‌های ناخواسته یا عنوان‌های تکراری را null کنیم. سپس از Fill Down استفاده می‌کنیم تا سلول‌های خالی با مقدار ردیف بالایی پر شوند. در پایان، فقط کافی است نام ستون‌ها را به شکل مناسب Rename کنیم.

File Home Transform Add Column View

Transpose Reverse Rows Count Rows

Data Type: Any Detect Data Type Rename

Replace Values Fill Pivot Column Convert to List

Unpivot Columns Move

Split Column Format ABC123 Extract Parse Merge Columns

Queries [7]

Challenge Challenge 2 مساله 2 تارگت ها تارگت (2) Sheet1 فرمت اصلی تار... Sheet1 (2)

fx = Table.ReorderColumns(#"Renamed Columns1",{"فصل", "ماه", "شرکت", "تارگت"})

	فصل	ماه	شرکت	تارگت
1	بهار	فروردین	گسترش داروی ایران	2000
2	بهار	فروردین	زیست دارو آوا	23000
3	بهار	فروردین	داروسازی نوین	900
4	بهار	اردیبهشت	گسترش داروی ایران	1590
5	بهار	اردیبهشت	زیست دارو آوا	3000
6	بهار	اردیبهشت	داروسازی نوین	2500
7	بهار	خرداد	گسترش داروی ایران	8000
8	بهار	خرداد	زیست دارو آوا	1200
9	بهار	خرداد	داروسازی نوین	2500
10	تابستان	تیر	گسترش داروی ایران	400
11	تابستان	تیر	زیست دارو آوا	1500
12	تابستان	تیر	داروسازی نوین	2000
13	تابستان	مرداد	گسترش دارو، ایران	1000

مسئله ۸ - ترکیب سطرها و ستون‌ها برای ساخت جدول تاریخ و مقدار

در این مسئله، جدول شامل چندین سطر و ستون مختلف است، اما هدف نهایی این است که فقط یک ستون تاریخ و ساعت داشته باشیم و در مقابل آن یک ستون مقدار (تعداد یا عدد).

توضیحات	هفته 1	هفته 2	هفته 3	هفته 4	هفته 5
1394	7/6/1394 12:00:00 AM	7/13/1394 12:00:00 AM	7/20/1394 12:00:00 AM	7/27/1394 12:00:00 AM	8/3/1394 12:00:00 AM
تعداد پرسنل فعال	247	241	242	240	242
1394	8/3/1394 12:00:00 AM	8/10/1394 12:00:00 AM	8/17/1394 12:00:00 AM	8/24/1394 12:00:00 AM	8/31/1394 12:00:00 AM
تعداد پرسنل فعال	238	231	231	231	230
1394	9/7/1394 12:00:00 AM	9/14/1394 12:00:00 AM	9/21/1394 12:00:00 AM	9/28/1394 12:00:00 AM	10/5/1394 12:00:00 AM
تعداد پرسنل فعال	227	231	228	227	226
1394	10/5/1394 12:00:00 AM	10/12/1394 12:00:00 AM	10/19/1394 12:00:00 AM	10/26/1394 12:00:00 AM	11/2/1394 12:00:00 AM
تعداد پرسنل فعال	229	227	227	229	229
1394	11/2/1394 12:00:00 AM	11/9/1394 12:00:00 AM	11/16/1394 12:00:00 AM	11/23/1394 12:00:00 AM	11/30/1394 12:00:00 AM
تعداد پرسنل فعال	226	223	223	228	237
1394	12/7/1394 12:00:00 AM	12/14/1394 12:00:00 AM	12/21/1394 12:00:00 AM	12/28/1394 12:00:00 AM	1/4/1395 12:00:00 AM
تعداد پرسنل فعال	242	238	245	242	240
1395	1/4/1395 12:00:00 AM	1/11/1395 12:00:00 AM	1/18/1395 12:00:00 AM	1/25/1395 12:00:00 AM	2/1/1395 12:00:00 AM
تعداد پرسنل فعال	230	238	239	240	239
1395	2/1/1395 12:00:00 AM	2/8/1395 12:00:00 AM	2/15/1395 12:00:00 AM	2/22/1395 12:00:00 AM	2/29/1395 12:00:00 AM
تعداد پرسنل فعال	242	238	245	242	242

ابتدا از مسیر **Home → Use First Row as Headers** استفاده می‌کنیم تا سطر اول به عنوان تیتر (Header) تبدیل شود. ستونی با عنوان «توضیحات» داریم که به آن نیاز نداریم. روی آن راست کلیک کرده و گزینه **Remove** را انتخاب می‌کنیم. سلول‌های خالی یا Null را باید با عدد صفر جایگزین کنیم. برای این کار از مسیر **Transform → Replace Values** استفاده می‌کنیم

Replace Values

Replace one value with another in the selected columns.

Value To Find: null

Replace With: 0

Advanced options

توضیحات	هفته 1	هفته 2	هفته 3	هفته 4	هفته 5
0	7/6/1394 12:00:00 AM	7/13/1394 12:00:00 AM	7/20/1394 12:00:00 AM	7/27/1394 12:00:00 AM	8/3/1394 12:00:00 AM
0	247	241	242	240	242
0	8/3/1394 12:00:00 AM	8/10/1394 12:00:00 AM	8/17/1394 12:00:00 AM	8/24/1394 12:00:00 AM	8/31/1394 12:00:00 AM
0	238	231	231	231	230
0	9/7/1394 12:00:00 AM	9/14/1394 12:00:00 AM	9/21/1394 12:00:00 AM	9/28/1394 12:00:00 AM	10/5/1394 12:00:00 AM
0	227	231	228	227	226
0	10/5/1394 12:00:00 AM	10/12/1394 12:00:00 AM	10/19/1394 12:00:00 AM	10/26/1394 12:00:00 AM	11/2/1394 12:00:00 AM
0	229	227	227	229	229
0	11/2/1394 12:00:00 AM	11/9/1394 12:00:00 AM	11/16/1394 12:00:00 AM	11/23/1394 12:00:00 AM	11/30/1394 12:00:00 AM
0	226	223	223	228	237
0	12/7/1394 12:00:00 AM	12/14/1394 12:00:00 AM	12/21/1394 12:00:00 AM	12/28/1394 12:00:00 AM	1/4/1395 12:00:00 AM
0	242	238	245	242	240
0	1/4/1395 12:00:00 AM	1/11/1395 12:00:00 AM	1/18/1395 12:00:00 AM	1/25/1395 12:00:00 AM	2/1/1395 12:00:00 AM
0	230	238	239	240	239
0	2/1/1395 12:00:00 AM	2/8/1395 12:00:00 AM	2/15/1395 12:00:00 AM	2/22/1395 12:00:00 AM	2/29/1395 12:00:00 AM
0	242	238	245	242	242
0	3/7/1395 12:00:00 AM	3/14/1395 12:00:00 AM	3/21/1395 12:00:00 AM	3/28/1395 12:00:00 AM	4/4/1395 12:00:00 AM
0	250	253	255	247	247
0	4/4/1395 12:00:00 AM	4/11/1395 12:00:00 AM	4/18/1395 12:00:00 AM	4/25/1395 12:00:00 AM	5/2/1395 12:00:00 AM
0	242	242	242	242	242

در این مرحله از Transform → Transpose استفاده می‌کنیم تا جدول بچرخد. به این ترتیب، ستون‌های تاریخ به ردیف تبدیل می‌شوند و ساختار جدول شبیه دو ستون می‌شود: یکی برای تاریخ و دیگری برای مقدار.

اکنون باید تمامی ستون‌ها را انتخاب کرده و از تب Transform → Unpivot Columns استفاده کنیم. با این کار، تمامی داده‌ها در دو ستون جدید قرار می‌گیرند.

Column1	Column2	Column3	Column4	Column5	Column6
7/6/1394 12:00:00 AM	247	8/3/1394 12:00:00 AM	238	9/7/1394 12:00:00 AM	227
7/13/1394 12:00:00 AM	242	8/10/1394 12:00:00 AM	231	9/14/1394 12:00:00 AM	228
7/20/1394 12:00:00 AM	241	8/17/1394 12:00:00 AM	231	9/21/1394 12:00:00 AM	231
7/27/1394 12:00:00 AM	240	8/24/1394 12:00:00 AM	225	9/28/1394 12:00:00 AM	230
0	0	8/31/1394 12:00:00 AM	230	0	0

در این مرحله باید از Add Column → Index Column و همچنین ستون‌های محاسباتی استفاده کنیم (مانند مساله شماره ۵)

Attribute	Value	Index	Integer-Division	Modulo
Column1	7/6/1394 12:00:00 AM	0	0	0
Column2	247	1	0	1
Column3	8/3/1394 12:00:00 AM	2	1	0
Column4	238	3	1	1
Column5	9/7/1394 12:00:00 AM	4	2	0
Column6	227	5	2	1
Column7	10/5/1394 12:00:00 AM	6	3	0
Column8	229	7	3	1
Column9	11/2/1394 12:00:00 AM	8	4	0
Column10	228	9	4	1
Column11	12/7/1394 12:00:00 AM	10	5	0
Column12	242	11	5	1
Column13	1/4/1395 12:00:00 AM	12	6	0
Column14	242	13	6	1
Column15	2/1/1395 12:00:00 AM	14	7	0
Column16	242	15	7	1
Column17	3/7/1395 12:00:00 AM	16	8	0

ستون Index را حذف می‌کنیم چون دیگر به آن نیاز نداریم، اکنون باید ستون Modulo را Pivot کنیم. در پایان فقط کافی است ستون‌ها را Rename کنیم، به این ترتیب جدول نهایی فقط دو ستون دارد.

تاریخ	پرسش فعال
7/6/1394 12:00:00 AM	247
8/3/1394 12:00:00 AM	238
9/7/1394 12:00:00 AM	227
10/5/1394 12:00:00 AM	229
11/2/1394 12:00:00 AM	228
12/7/1394 12:00:00 AM	242
1/4/1395 12:00:00 AM	242
2/1/1395 12:00:00 AM	242
3/7/1395 12:00:00 AM	250
4/4/1395 12:00:00 AM	245
5/2/1395 12:00:00 AM	242
6/6/1395 12:00:00 AM	236
7/4/1395 12:00:00 AM	242
8/1/1395 12:00:00 AM	234
9/5/1395 12:00:00 AM	233
10/3/1395 12:00:00 AM	238
11/7/1395 12:00:00 AM	236
12/5/1395 12:00:00 AM	241
1/2/1396 12:00:00 AM	227
2/6/1396 12:00:00 AM	236

جلسه دهم

زبان M

زبان فرمول نویسی و پشت صحنه Power Query است. این زبان عمدتاً functional است و با نوشتن functions می توان به خواسته های پاک سازی و تغییر شکلی داده ها و جدول ها رسید.

آشنایی با محیط کدنویسی

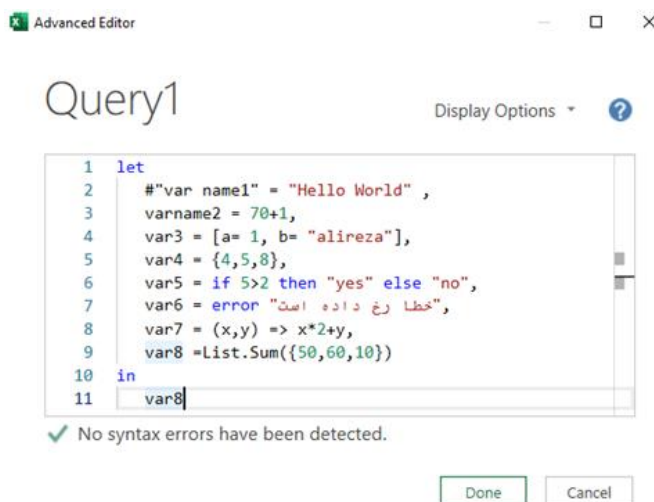
در M معمولاً از ساختار زیر استفاده می کنیم: ابتدا متغیرها را با let تعریف می کنیم، سپس محاسبات را انجام می دهیم و در انتها با in خروجی را مشخص می کنیم. بین دستورات داخل let باید کاما (,) گذاشته شود؛ ولی قبل از in نباید کاما قرار گیرد.

نکات عملی در محیط کدنویسی:

- یک کوئری خالی ایجاد می کنیم و در Advanced Editor وارد محیط کدنویسی می شویم.
- از بخش Display Options گزینه هایی را که نیاز داریم فعال می کنیم تا نمایش مناسبی داشته باشیم.
- اگر اسم متغیر یا نام ستون شامل فاصله باشد (مثلاً First Name)، باید آن را داخل دابل کوتیشن قرار دهیم و

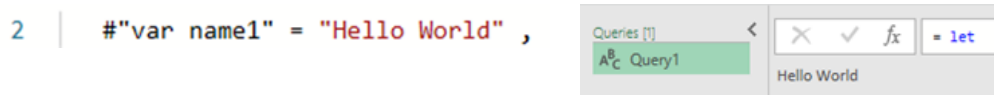
قبلش علامت # اضافه کنیم: مثال. "#First Name".

عبارات در Power Query (M language) چند نوع هستند. در ادامه چند نمونه از انواع داده و ساختارهای پایه را با مثال آورده ام:



- A text value

یک مقدار متنی که داخل double quotation (") قرار می گیرد.



• A number

یک عدد ساده — می تواند صحیح یا اعشاری باشد.

```
2 | var2 = 50
```

Queries [1]	<	X	✓	fx	= 50
1 ² Query1					50

• A sum of two numbers

یک عبارت عددی که نتیجه جمع دو عدد است.

```
3 | varname2 = 70+1,
```

Queries [1]	<	X	✓	fx	= let
1 ² Query1					71

• A list of three numbers

یک List مجموعه ای از مقادیر است که عناصرش با کاما جدا می شوند.

```
5 | var4 = {4,5,8},
```

Queries [1]	<	X	✓	fx	= let
1 ² Query1					list
					1 4
					2 5
					3 8

• A record containing two fields

Record نوعی از داده است که داخل آن ساختارمند نیست و می توان ساختار آن را تعریف کرد.

```
4 | var3 = [a= 1, b= "alireza"],
```

Queries [1]	<	X	✓	fx	= let
1 ² Query1					a 1
					b alireza

• A computed expression

عبارتی است که حاصل یک محاسبه یا ترکیب مقادیر مختلف است. مثلاً وقتی از علامت جمع (+) یا عملگرهای دیگر

استفاده می کنیم .

```
8 | var7 = (x,y) => x*2+y,
```

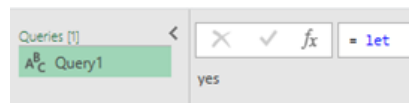
```
10 | in
11 | var7(2,1)
```

Queries [1]	<	X	✓	fx	= let
1 ² Query1					5

• A conditional expression

یک عبارتی شرطی است.

```
6 | var5 = if 5>2 then "yes" else "no",
```



• A let expression

این نوع عبارت برای تعریف متغیرها و سازماندهی کد استفاده می‌شود.

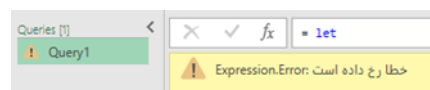
```
3 | var9 = 1+2
4 | in
5 | var9*2
```



• An error with message

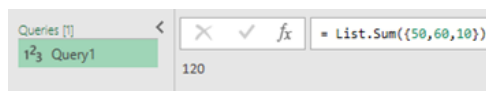
برای تولید خطا (Error) همراه با پیام دلخواه از تابع error استفاده می‌شود.

```
7 | var6 = error "خطا رخ داده است",
```



تابع List.Sum یکی از توابع ثابت (built-in) زبان M است.

```
9 | var8 =List.Sum({50,60,10})
```

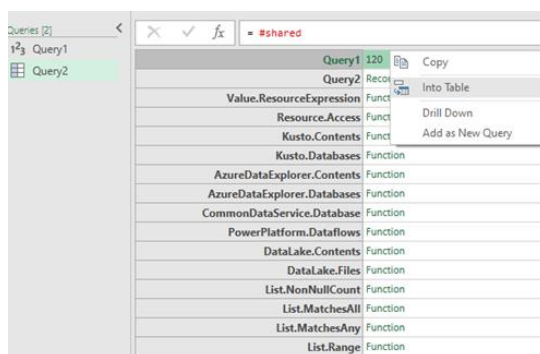


شماره	نوع Expression	مثال	توضیح
۱	Text	"Hello World"	متن درون کوتیشن
۲	Number	123	عدد صحیح یا اعشاری
۳	Sum Numbers	1 + 2	عبارت ریاضی
۴	List of Numbers	{1, 2, 3}	مجموعه‌ای از داده‌ها
۵	Record	[X=1, Y=2+3]	ساختار کلید-مقدار
۶	Computed Expression	1 + 2 + 3	انجام محاسبه
۷	Conditional Expression	if x > 1 then "Yes" else "No"	شرطی
۸	Let Expression	let a=1,b=2 in a+b	تعریف متغیرها و خروجی
۹	Error Expression	Error "خطا رخ داده است"	ایجاد خطا با پیام

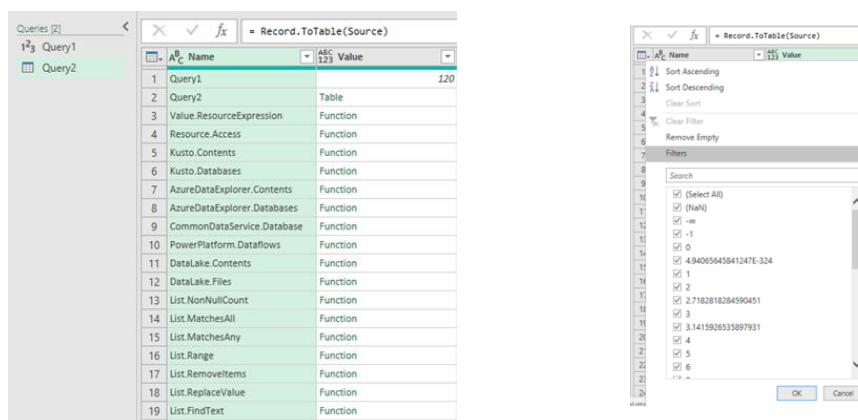
مشاهده و شناسایی توابع در (Power Query (M Language

در زبان M ، همه ی توابع پیش فرض نگهداری می شوند. برای دیدن لیست تمام آنها می توانیم در یک New Query در قسمت فرمول #shared= بنویسیم، این دستور یک لیست از تمام آبجکت هایی که در Power Query در دسترس هستند را نمایش می دهد، توابع مقادیر، جداول و کوئری هایی که خودمان ایجاد کردیم.

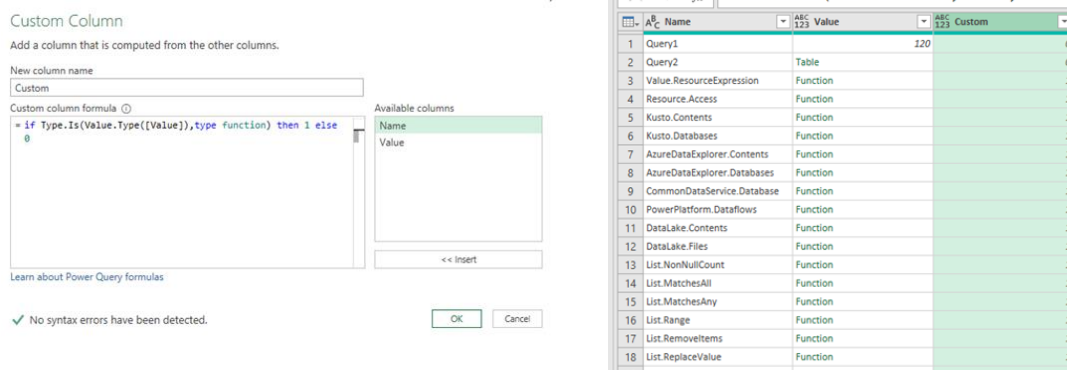
برای اینکه بتوانیم داده های داخل #shared رو بررسی کنیم، روی خروجی List راست کلیک و گزینه ی To Table را انتخاب می کنیم.



حالا داده ها در قالب یک جدول دو ستونه دیده می شوند.

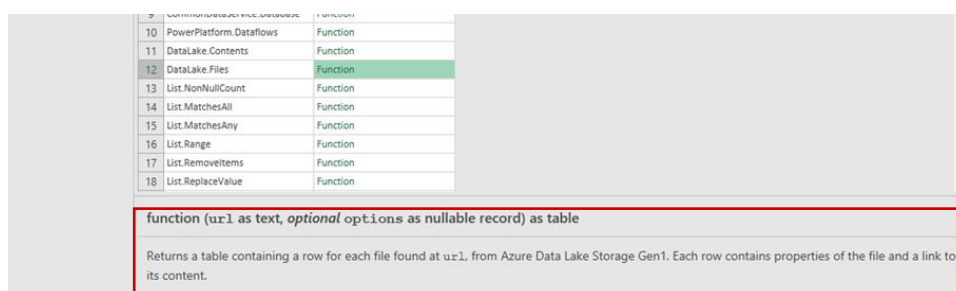


اما توابع دیده نمی شوند، در Power Query نوع خاصی از داده به نام function type است و برخلاف متن یا عدد، این نوع مستقیماً در جدول قابل فیلتر نیست و نمایش داده نمی شوند. برای تشخیص اینکه کدام ردیف تابع است، در Add Column → Custom Column باید دستور زیر را بنویسیم



در واقع در این دستور می‌خواهیم بدانیم که در لیست هر جا Function بود در ستون جدید عدد ۱ را برگرداند و در غیراینصورت عدد صفر را.

بعد از ساخت ستون جدید، روی ستون فیلتر می‌گذاریم تا فقط ردیف‌هایی که مقدارش 1 هست نمایش داده بشوند. حالا فقط توابع در لیست باقی می‌مانند. وقتی روی هر تابع کلیک کنیم، در پایین پنجره‌ی Power Query توضیح مختصری از آن تابع نمایش داده می‌شود — شامل نوع ورودی‌ها، خروجی، و هدف تابع.



توابع Power Query در حدود ۲۴ گروه اصلی طبقه‌بندی شده‌اند.

- | | |
|-----------------------------|------------------------|
| 1. Accessing data functions | 14. Logical functions |
| 2. Binary functions | 15. Number functions |
| 3. Combiner functions | 16. Record functions |
| 4. Comparer functions | 17. Replacer functions |
| 5. Date functions | 18. Splitter functions |
| 6. DateTime functions | 19. Table functions |
| 7. DateTimeZone functions | 20. Text functions |
| 8. Duration functions | 21. Time functions |
| 9. Error handling | 22. Type functions |
| 10. Expression functions | 23. Uri functions |
| 11. Function values | 24. Value functions |
| 12. List functions | |
| 13. Lines functions | |

مقایسه تعداد توابع در زبان‌های داده‌ای

نام قابلیت یا نرم افزار	کاربرد نرم افزار	تعداد توابع
Excel Functions	تحلیل و محاسبات فردی روی داده‌های جدولی	+۵۰۰
Power Query (M Language)	آماده‌سازی و (Etl) استخراج، تبدیل، بارگذاری داده	+۷۰۰
DAX Functions	مدل‌سازی و محاسبات تحلیلی در Power BI و ...	+۳۵۰
SQL Functions	پرس‌وجو، مدیریت و پردازش داده در دیتابیس‌ها	+۳۰۰

این تفاوت اعداد نشان می‌دهد که زبان M در بخش تبدیل (Transformation) و پاک‌سازی داده‌ها قابلیت بسیار بیشتری دارد،

منابع یادگیری زبان M

برای یادگیری و جستجوی توابع عبارت Power Query M Language را جستجو کنید،

[/https://learn.microsoft.com/en-us/powerquery-m](https://learn.microsoft.com/en-us/powerquery-m)

[/https://powerquery.how](https://powerquery.how)

در این سایت‌ها می‌توانید تمام توابع M را ببینید، گروه‌بندی (category) آن‌ها را بررسی کنید مثال و توضیح هر تابع را بخوانید.

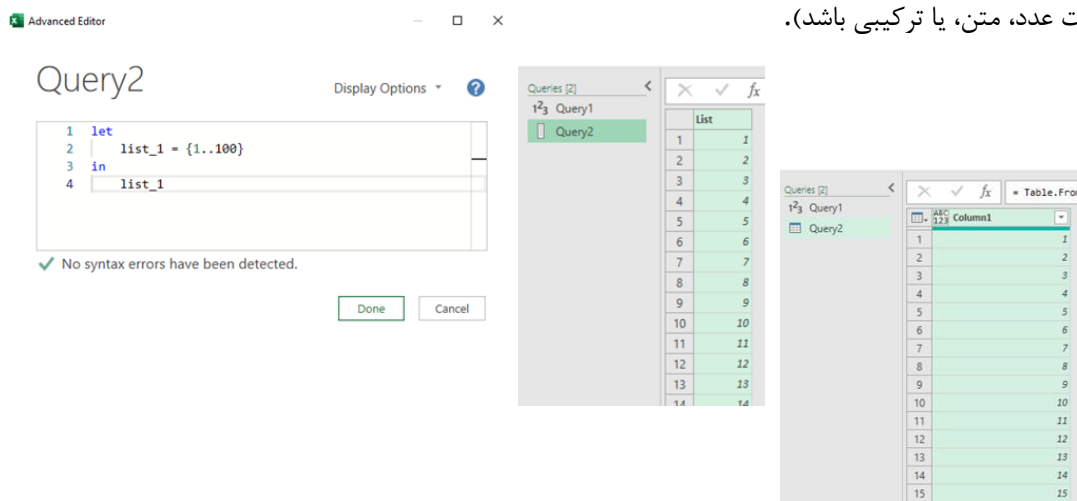
دلایل استفاده از زبان M

دلیل اول: کند شدن فایل‌های سنگین

ممکن است داده‌های زیادی وجود داشته باشد و محاسبات پیچیده باشند و پردازش در Power Query سنگین و کند شود. در این مواقع باید از یک سری functions استفاده کرد تا جلوی کند شدن کار گرفته شود.

مثال: ایجاد لیست و تبدیل آن به جدول

یک کوئری جدید می‌سازیم و در محیط کدنویسی لیست خود را ایجاد می‌کنیم. سپس گزینه To Table می‌زنیم، خروجی، جدولی با یک ستون است که اعداد ۱ تا ۱۰۰ در آن نمایش داده می‌شوند، اما نوع داده در این حالت Any است (ممکن است عدد، متن، یا ترکیبی باشد).



برای اینکه داده‌ها نوع مشخصی داشته باشند، باید قبل از ساخت جدول آن‌ها را تبدیل کنیم



- List.Transform : روی هر آیتم از لیست، تابعی اعمال می‌کند.
- each Text.From(_): یعنی هر مقدار را به متن تبدیل کن.
- "Number" اسم ستون جدول است.

اگر بخواهیم ستون را از Text به Number برگردانیم:

Query2

Display Options ?

```
2 list_1 = {1..100},
3 result =Table.FromList(List.Transform(list_1,each Text.From(_)),null,{"numbers"}),
4 result2 =Table.TransformColumnTypes(result,{"numbers",Int64.Type})
5
6 in
7 result2
```

1 ² numbers
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

حالا می‌خواهیم لیست جدیدی ایجاد کنیم که ۱۰۰۰۰ سطر دارد و در ستون بعدی هم جمع تجمعی این اعداد را می‌خواهیم.

• روش اولیه (روش دستی و کند)

ابتدا در Power Query از منوی 1 → From Index Column → Add Column استفاده می‌کنیم. ستونی به

نام مثلاً Index اضافه می‌شود. سپس از Add Column → Custom Column یک ستون جدید می‌سازیم و کد زیر را می‌نویسیم:

1 ² numbers	1 ² Index
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19

Custom Column

Add a column that is computed from the other columns.

New column name: Custom

Custom column formula: = List.Sum(List.FirstN(result2[numbers],[Index]))

Available columns: numbers, Index

<< Insert

OK Cancel

تابع List.FirstN: به هر سطر که می‌رسد n تا اول آن سطر می‌شود اعدادی که باید جمع زده شوند. حالا یک

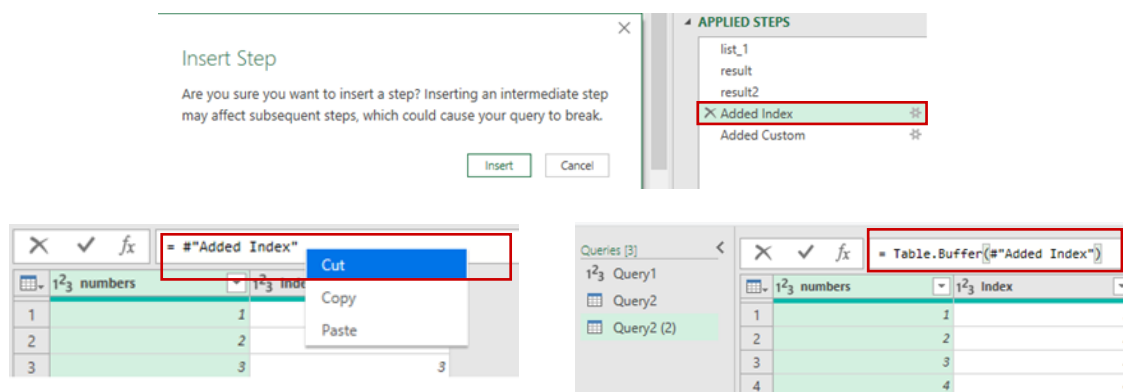
ستون تجمعی بوجود می‌آید و اگر فایل را در اکسل بارگذاری کنیم، بسیار زمانبر خواهد شد و کند عمل خواهد کرد.

• روش بهینه با استفاده از تابع Table.Buffer

برای رفع کندی تا داده‌ها فقط یک بار در حافظه (RAM) بارگذاری شوند.

مراحل: در Power Query روی گام قبل از Custom Column مثلاً "#Added Index" کلیک می‌کنیم. روی نوار فرمول (fx) کلیک می‌کنیم تا یک گام جدید ایجاد شود. نام گام قبلی را کات کرده و، از تابع Table.Buffer استفاده می‌کنیم

= Table.Buffer("#Added Index")



این کار باعث می‌شود که جدول "Added Index" یک بار در حافظه ذخیره شود و دیگر در هر سطر مجدد خوانده نشود. در نتیجه، ستون تجمعی همان خروجی قبلی را دارد، اما زمان بارگذاری حدود ۳۰ تا ۴۰ درصد کمتر می‌شود. (چون Table.Buffer داده را در حافظه موقت نگه می‌دارد و از خواندن مجدد منبع جلوگیری می‌کند.)

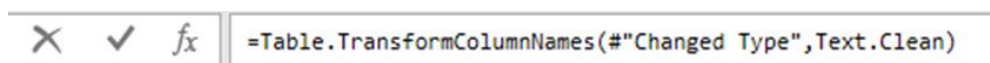
دلیل دوم: پاک‌سازی سرستون‌های یک (Header) کثیف در Power Query

گاهی نام ستون‌ها در داده‌ها نامرتب، دارای فاصله، کاراکتر خاص، یا حتی زبان ترکیبی (مثلاً عربی و فارسی) هستند. برای اصلاح همه سرستون‌ها بصورت خودکار، از تابع زیر استفاده می‌کنیم:

Table.TransformColumnNames

برای مثال جدولی داریم که در سرستون‌ها حروفی مانند "ی" باید یکسان شوند، در fx کلیک کرده و تابع موردنظر را

می‌نویسیم.



تابع Text.Clean یک ورودی دوم هم دارد، که می‌توانیم از هوش مصنوعی کمک بگیریم و کد مرحله‌ی قبلی را در ChatGPT کپی کنیم و بگوییم:

ChatGPT ▾

= Table.TransformColumnNames("#Changed Type",Text.Clean)
میخواهم در ورودی دوم این تابع کدی بنویسم که برای تمام ستونها "ی" عربی را با "ی" فارسی جایگزین کند، چه باید بنویسم؟

+

🗣

↑

پاسخ این کد است که آن را کپی نموده و در فرمول اضافه می کنیم و سرستون ها اصلاح می گردد.

Copy code

```
= Table.TransformColumnNames("#Changed Type", each Text.Replace(_, "ی", "ی"))
```

×	✓	f _x	= Table.TransformColumnNames("#Changed Type", each Text.Replace(_, "ی", "ی"))	▾
📅	📄	📊	📌	📌
فاکتور و 2	نوع فاکتور	مرکز مشتری	نام مشتری	کد مشتری و 2
گروه فروش	گروه فروش	گروه فروش	گروه فروش	گروه فروش

دلیل سوم: فیلتر بدون حساسیت به حروف بزرگ/کوچک

به صورت پیش فرض، Power Query در فیلتر کردن به حروف حساس است. (Case-Sensitive) اما می توانیم این حساسیت را با پارامتر سوم تابع Table.SelectRows کنترل کنیم.

برای مثال که اگر بخواهیم کلمه ای را فیلتر کنیم براساس اینکه با حروف بزرگ یا کوچک فیلتر شود، فقط همان کالا را برمی گرداند، اما ما می خواهیم لیست تمام آن کالا را برای ما برگرداند و غیر حساس به حروف باشد، از مقایسه کننده ی زیر استفاده می کنیم:

Comparer.OrdinalIgnoreCase

مثال ۱: فیلتر نام کالا بدون حساسیت به حروف

فرض کنیم جدولی داریم که در ستون "نام کالا" می خواهیم همه کالاهایی که نامشان شامل "xerox" است را ببینیم — چه با X بزرگ چه کوچک (درواقع به حروف بزرگ و کوچک حساس نباشد) و اگر از فیلتر استفاده کنیم همه کالاها را بر نمی گرداند، پس باید تابعی زیر را بنویسیم :

×	✓	f _x	= Table.SelectRows("#Changed Type", each Text.Contains([نام کالا], "Xerox", Comparer.OrdinalIgnoreCase))	▾
📅	📄	📊	📌	📌
فاکتور و 2	نوع فاکتور	مرکز مشتری	نام مشتری	کد مشتری و 2
گروه فروش	گروه فروش	گروه فروش	گروه فروش	گروه فروش

ردیف	کد کالا	گروه کالا	زیر گروه کالا	نام کالا
1	OFF-PA-10002365	Office Supplies	Paper	Xerox 1967
2	OFF-PA-10001569	Office Supplies	Paper	Xerox 232
3	OFF-PA-10003892	Office Supplies	Paper	Xerox 1943
4	OFF-PA-10000904	Office Supplies	Paper	xerox 232
5	OFF-PA-10003177	Office Supplies	Paper	Xerox 1999
6	OFF-PA-10004965	Office Supplies	Paper	xerox 1921
7	OFF-PA-10002377	Office Supplies	Paper	Xerox 1916
8	OFF-PA-10001804	Office Supplies	Paper	Xerox 195
9	OFF-PA-10001736	Office Supplies	Paper	Xerox 1880
10	OFF-PA-10004327	Office Supplies	Paper	Xerox 1911
11	OFF-PA-10002254	Office Supplies	Paper	Xerox 1883
12	OFF-PA-10002751	Office Supplies	Paper	Xerox 1920
13	OFF-PA-10001144	Office Supplies	Paper	Xerox 1913
14	OFF-PA-10000061	Office Supplies	Paper	xerox 205
15	OFF-PA-10002479	Office Supplies	Paper	Xerox 4200 Series MultiUse Premium Copy Paper (20Lb. and 84 Bright)

دلیل چهارم: خودکارسازی فرایندهای پیچیده

مثال Import Folder: که فایل‌ها ترکیبی از csv و Excel هستند. وقتی داده‌ها وارد فولدر می‌شوند، می‌توانیم با M Language تمام فایل‌ها را خودکار وارد و پردازش کنیم. بدون نیاز به انجام دستی هر بار، Power Query مراحل را تکرار و داده‌ها را یکپارچه می‌کند.

دلیل پنجم: ایجاد جدول سفارشی (Custom Table)

مثال: تولید لیست تاریخ‌ها با استفاده از List.Generate و تبدیل آن به جدول.
توضیح:

- List.Generate → تولید یک لیست پویا با شرط و افزایش مشخص
- Date.AddDays(_,1) → اضافه کردن یک روز به هر آیت

می‌توانیم به جای ۱ روز، عدد دلخواهی (مثلاً ۲ یا ۳) اضافه کنیم تا فاصله‌ها تغییر کند.

The screenshot shows the Advanced Editor with a query named 'Query1'. The M code is as follows:

```
1 let
2     mycal = List.Generate(
3         () => #date(2020,1,1),
4         each _ <= #date(2030,12,1),
5         each Date.AddDays(_,1),
6         each _
7     )
8 in
9     mycal
```

Below the code, it says 'No syntax errors have been detected.' There are 'Done' and 'Cancel' buttons. To the right, a preview of the resulting table is shown, containing a list of dates from 1/1/2020 to 1/18/2020.

دلیل ششم: تغییر و تبدیل نوع داده‌ها

گاهی در فرمول‌ها، داده‌ها از حالت عدد (Number) به متن (Text) یا برعکس تغییر می‌کنند

The screenshot shows a table with columns: 'نام', 'نوع ارسال', 'تاریخ ارسال', 'تاریخ سفارش', 'شماره سفارش', 'فروش تعدادی و؟', 'فروش دلاری 1.2', 'تحقیق 1.2', 'سود 1.2'. The first two rows of data are highlighted. The third column, 'فروش تعدادی و؟', contains the values '379.372' and '46.2', both followed by 'Error' in red text, indicating a data type mismatch.

ستونی که دارای Error می‌باشد را پیدا کرده و در فرمول آن نوع داده را از Int64.Type به type text تغییر می‌دهیم.


```
= Table.TransformColumnTypes("#Promoted Headers",{{"شماره سفارش", type text}, {"تاریخ سفارش", type date}, {"تاریخ ارسال", type date}, {"نوع ارسال", type text}, {"نام مشتری", type text}, {"نوع مشتری", type text}, {"کشور", type text}, {"شهر", type text}, {"زیر گروه", type text}, {"گروه کالا", type text}, {"کد کالا", type text}, {"ناحیه", type text}, {"ایالت", type text}, {"نام کالا", type text}, {"تاریخ سفارش", type number}, {"تاریخ ارسال", type number}, {"تاریخ سفارش", type number}, {"تاریخ ارسال", type number}}})
```

تاریخ سفارش	تاریخ ارسال	نوع ارسال	نام مشتری	نوع مشتری
11/8/2016	11/11/2016	Second Class	Claire Gute	Consumer
11/8/2016	11/11/2016	Second Class	Claire Gute	Consumer
6/12/2016	6/16/2016	Second Class	Darrin Van Huff	Corporate
10/11/2015	10/18/2015	Standard Class	Sean O'Donnell	Consumer
10/11/2015	10/18/2015	Standard Class	Sean O'Donnell	Consumer

```
= Table.TransformColumnTypes("#Promoted Headers",{{"شماره سفارش", type text}, {"تاریخ سفارش", type date}, {"تاریخ ارسال", type date}, {"نوع ارسال", type text}, {"نام مشتری", type text}, {"نوع مشتری", type text}, {"کشور", type text}, {"شهر", type text}, {"زیر گروه", type text}, {"گروه کالا", type text}, {"کد کالا", type text}, {"ناحیه", type text}, {"ایالت", type text}, {"نام کالا", type text}, {"تاریخ سفارش", type number}, {"تاریخ ارسال", type number}, {"تاریخ سفارش", type number}, {"تاریخ ارسال", type number}}})
```

دلیل هفتم: افزودن دیتا در کدنویسی (Append Data / Insert Row)

در پاور کوئری و محیط اکسل می‌توان دیتا ایجاد و سپس به جدول اصلی Append نمود. در Power bi هم می‌توان Inter Data کرد و بعد آن را Append کرد، اما گاهی نیاز داریم یک سطر داده فقط به جدول اضافه شود و در جای دیگری اطلاعات ثبت نشود، برای مثال فایل اصلی داده ناقص است (مثلاً یک فاکتور در سایت موجود نیست) می‌خواهیم با کدنویسی یک سطر به جدول اضافه کنیم.

ابتدا جدول را در محیط اکسل بارگذاری کرده، سرستون‌ها را کپی و در ChatGPT آنها را اضافه می‌کنیم و دستوری برای اضافه شدن یک ردیف دیتا با این سرستون‌ها را درخواست می‌کنیم.

ChatGPT ▾

order_id order_date order_price order_qty customer_id
یک سطر داده باید به جدول فوق توسط پاور کوئری اضافه کنم حاوی دیتای زیر
1000 1404/02/15 500 2 60

Sheet1 (2)

Display Options ▾ ?

```
1 let
2 Source = Excel.Workbook(File.Contents("D:\Power_Query\10\data\07_فاکتورهای_صایت.xlsx"), null, true),
3 Sheet1_Sheet = Source[[Item="Sheet1",Kind="Sheet"]][Data],
4 #"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet, [PromoteAllScalars=true]),
5 #"Changed Type" = Table.TransformColumnTypes("#Promoted Headers",{{"order_id", Int64.Type}, {"order_date", type text}, {"order_price", Int64.Type}, {"order_qty", Int64.Type}, {"customer_id", Int64.Type}}),
6 NewRow = {[order_id = 1000, order_date = "1404/05/01", order_price = 500, order_qty = 2, customer_id = 60]},
7
8 // اضافه کردن سطر جدید به جدول
9 AddedRow = Table.InsertRows("#Changed Type", Table.RowCount("#Changed Type"), NewRow)
10 in
11 AddedRow
```

خروجی در جدول و سطر آخر اضافه خواهد شد.



	1 ² order_id	A ⁸ order_date	1 ² order_price	1 ² order_qty	1 ² customer_id
1	1005	1404/5/11	2560000	2	114
2	1006	1404/5/11	1900000	2	156
3	1007	1404/5/11	2560000	1	173
4	1008	1404/5/11	1900000	1	182
5	1009	1404/5/13	2990000	1	193
6	1010	1404/5/14	2990000	1	145
7	1011	1404/5/14	1900000	2	179
8	1012	1404/5/15	2990000	1	158
9	1013	1404/5/15	2990000	2	149
10	1014	1404/5/19	1900000	1	137
11	1015	1404/5/22	2560000	1	154
12	1016	1404/5/22	2560000	2	116
13	1017	1404/5/27	2990000	2	148
14	1018	1404/5/29	1900000	2	105
15	1019	1404/5/3	2560000	2	174
16	1020	1404/5/3	1900000	1	102
17	1021	1404/5/30	2990000	2	102
18	1022	1404/5/7	1900000	1	146
19	1023	1404/5/8	2990000	2	180
20	1000	1404/05/01	500	2	60

لینک سایت جهت مشاهده دوره آموزشی:

/ <https://www.tahlilgari.com/pgfree>