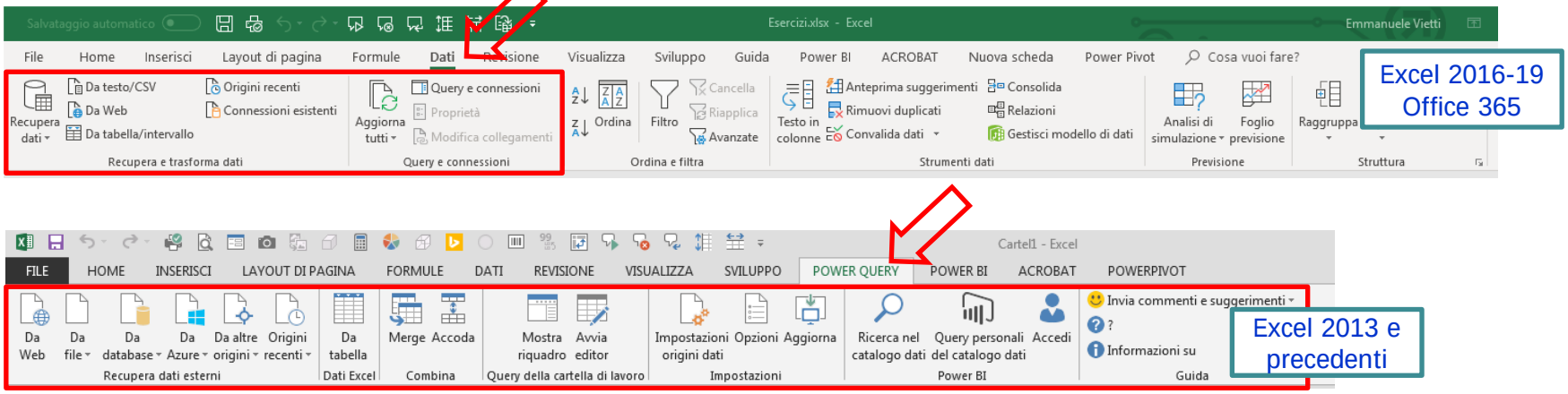
Utilizzo avanzato di Microsoft Excel

Ms Excel: Power Query: nuovi strumenti per l'acquisizione e l'elaborazione dei dati

INTRODUZIONE A POWER QUERY

COSA E' POWER QUERY?

- Power Query nasce come un componente aggiuntivo di MS-Excel che facilita l'importazione e la manipolazione di dati dalle più svariate fonti: si può considerare una "super versione" del solito "Carica dati esterni" presente nel menù/scheda "Dati" delle varie versioni di Excel.
- Una volta installato presenta una propria scheda in Excel 2010 e 2013, mentre in Excel 2016 è integrato (nativo) in **Dati > Recupera e trasforma**
- In Office 2010 e 2013 Power Query deve essere attivato (a seconda della versione potrebbe essere necessario scaricare un add-in: http://www.microsoft.com/it-it/download/details.aspx?id=39379) attraverso **File > Opzioni > Componenti aggiuntivi COM > Microsoft Power Query per Excel**



COSA E' POWER QUERY?

- Power Query è parte della suite Power BI per Office 365 (e ne costituisce il motore ETL):

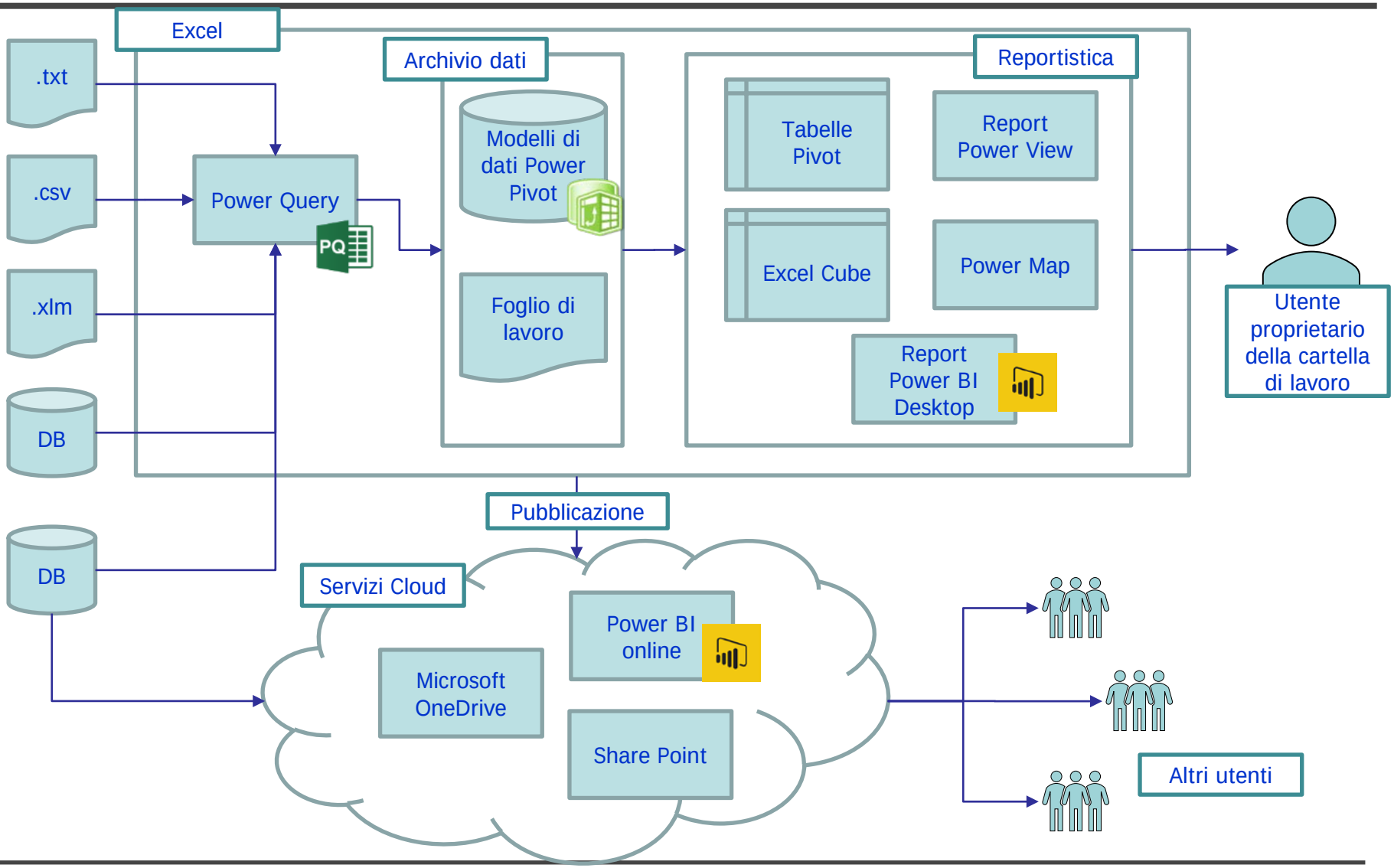
  Ogni strumento di Business Intelligence ha tre funzioni principali:

  - Importazione e trasformazione dei dati (ETL) → Power Query
  - Strutturazione delle reazione tra dati (OLAP) → Power Pivot
  - Reporting → Power View / Power Map / PowerBI Desktop

  Power BI comprende anche servizi cloud per la condivisione dei report.

- Power Query è il primo componente della suite Power BI che si incontra quando si vuole strutturare una nuova soluzione di BI (ma non necessariamente solo per questo). Power Query permette di:
  - Connettersi da un'ampia varietà di fonti
  - Estrarre dati in modo semplice e veloce
  - Definire una serie di «step» (passi) ripetibili per ripulire, filtrare … trasformare la basedati prima di caricarlo su Excel (sia esso il foglio di lavoro o, dalla versione 2013, nei modelli di dati di Excel / Power Pivot)

Ms Excel: Power Query: nuovi strumenti per l'acquisizione e l'elaborazione dei dati



COSA E' UN MODELLO DI DATI DI EXCEL?

- Il modello dati di Excel (disponibile nativamente dalla versione 2013) è un motore di database in memoria attivo nell'applicazione che permette di caricare e elaborare grandi quantità di dati.
- Introdotto con Power Pivot come add-in nella versione 2010 (che comprendeva database e interfaccia utente). Dalla versione 2013 il database (modelli di dati) è stato integrato direttamente in Excel e Power Pivot continua ad essere un componente aggiuntivo, comprendendo solo l'interfaccia utente.
- Quali sono i vantaggi di lavorare con i modelli dei dati:
  - Possibilità di lavorare con grandi quantità di dati (oltre il milione di righe presente sul foglio di Excel – fino anche a centinaia di milioni di righe: non ci sono limiti, dipende dal tipo di dati e dalla potenza del computer)
  - Database su «fogli» nascosti
  - Possibilità di caricare più tabelle dati e creare relazioni
  - Effettuare calcoli complessi, più velocemente, attraverso l'utilizzo del linguaggio DAX
  - Compressione dei dati molto più efficiente (con relativa riduzione della dimensione della cartella di lavoro)
  - E' possibile creare gerarchie che permetto all'utente di utilizzare la funzione di Drill Down dei dati

COSA E' UNA QUERY?

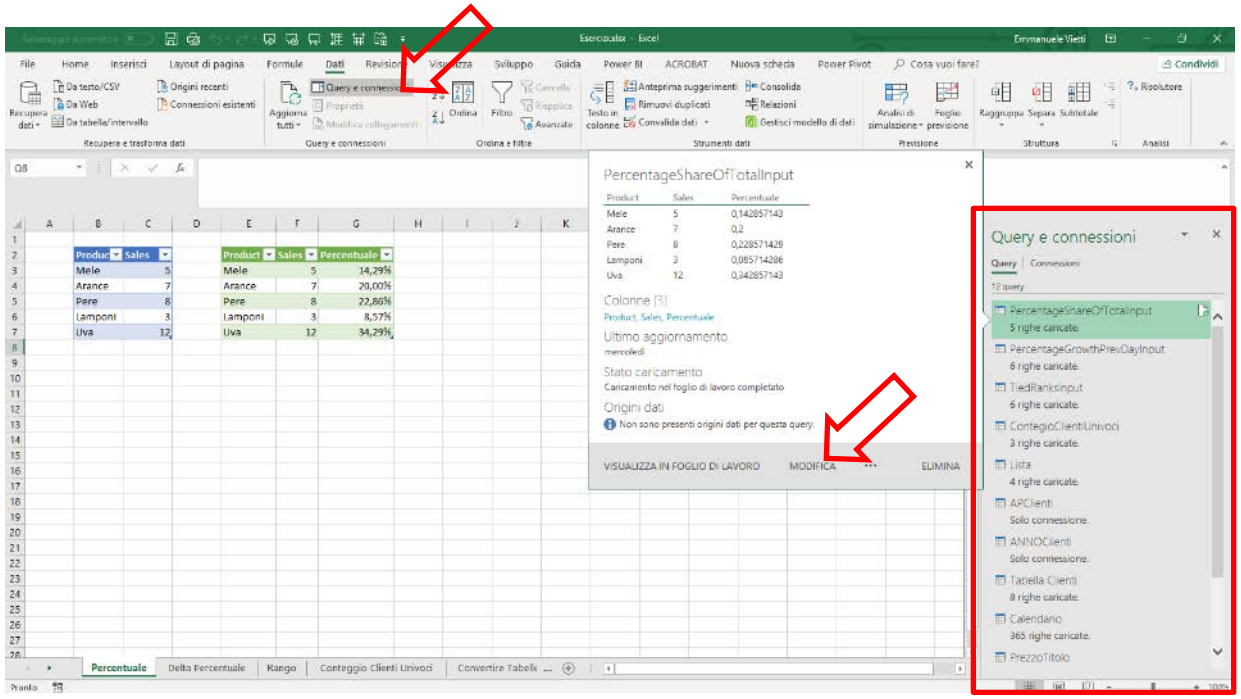
- Il concetto più importante di Power Query è quello di **query** (e purtroppo si parlerà delle Query di Power Query ☺).

Query (s.m.): In informatica, interrogazione di un database per estrarre o aggiornare i dati che soddisfano un certo criterio di ricerca.

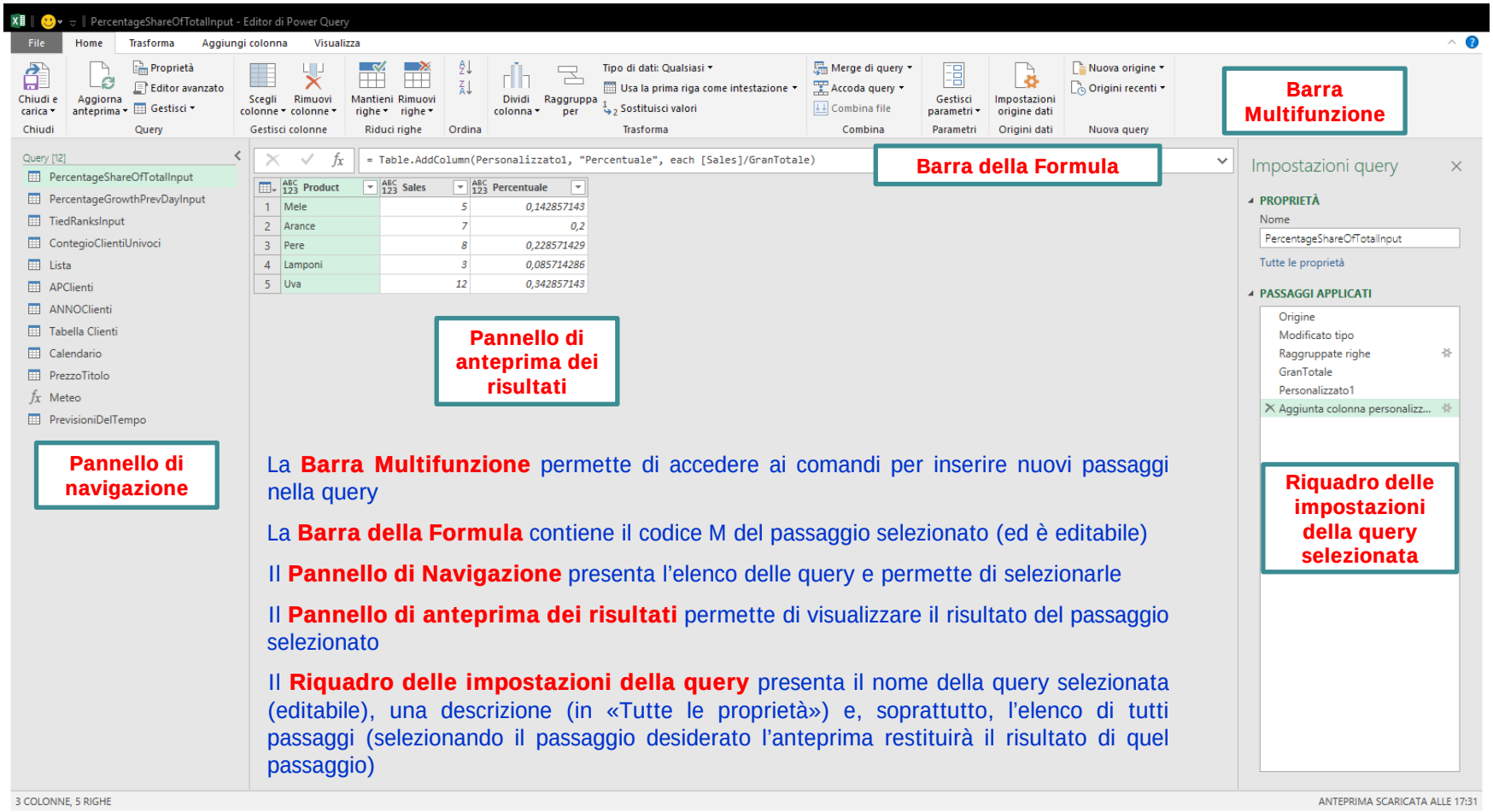
- Una query è costituita da una serie di passaggi (step o passi) che permettono di acquisire dati da una o più fonti, opzionalmente li trasforma (es: filtra, aggrega, etc.) e successivamente li carica in Excel.
- Le query sono salvate nella cartella di lavoro. E ogni cartella di lavoro può contenere più query.
- Le query possono a loro volta rappresentare la fonte dati di altre query.
- I diversi passaggi in Power Query sono codificati con un linguaggio nativo (linguaggio M) che purtroppo non ha alcuna similarità né con le classiche formule di Excel né con VBA. Tuttavia nella maggior parte dei casi non sarà necessario codificare la query, in quanto il punto di forza di Power Query è la facilità di utilizzo dell'interfaccia utente che permette di ovviare la stesura del codice.

COME SI ACCEDE A POWER QUERY?

- Se la cartella di lavoro presenta già delle query occorre attivare il riquadro Query e Connessioni (**Dati > Query e connessioni >Query e connessioni**) posizionarsi con il mouse su una query e **cliccare modifica** nella finestra di dialogo che comparirà (oppure **pulsante dx del mouse > Modifica**)
- Se la cartella di lavoro non presenta delle query occorre procedere con la selezione di una fonte dati (**Dati > Recupera e trasforma dati > …**) e si accederà all'interfaccia utente di Power Query, oppure **Dati > Recupera e trasforma dati > Recupera Dati > Avvia l'Editor di Power Query**



L'INTERFACCIA UTENTE: EDITOR DI POWER QUERY



La **Barra Multifunzione** permette di accedere ai comandi per inserire nuovi passaggi nella query

La **Barra della Formula** contiene il codice M del passaggio selezionato (ed è editabile)

Il **Pannello di Navigazione** presenta l'elenco delle query e permette di selezionarle

Il **Pannello di anteprima dei risultati** permette di visualizzare il risultato del passaggio selezionato

Il **Riquadro delle impostazioni della query** presenta il nome della query selezionata (editabile), una descrizione (in «Tutte le proprietà») e, soprattutto, l'elenco di tutti passaggi (selezionando il passaggio desiderato l'anteprima restituirà il risultato di quel passaggio)

PERCHE' USARE POWER QUERY?

- Power Query è il prodotto su cui si sono concentrati gli sviluppi degli ultimi anni, in Excel 2016 Office 365 è diventato l'unico strumento per acquisire e rielaborare dati esterni (Importa dati è stato dismesso).
- Le fonti dati supportate sono numerosissime (inclusi i sevizi web)
- Possibilità di automatizzare numerosi passaggi manuali, con conseguente risparmio di tempo
- Estrema facilità di trasformare i dati e aggiungere calcoli (non tutto si può fare con l'interfaccia utente ma gran parte si)
- Elaborare i dati sulla base dati e non sulla cartella di lavoro
- Possibilità di estrarre e condividere le query con altri utenti

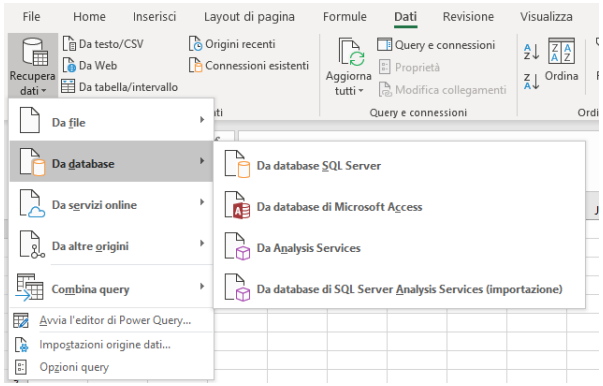
Guarda il video tutorial di introduzione su YouTube:
EXCEL: INTRODUZIONE A POWER QUERY - DIRETTA #10 (link)

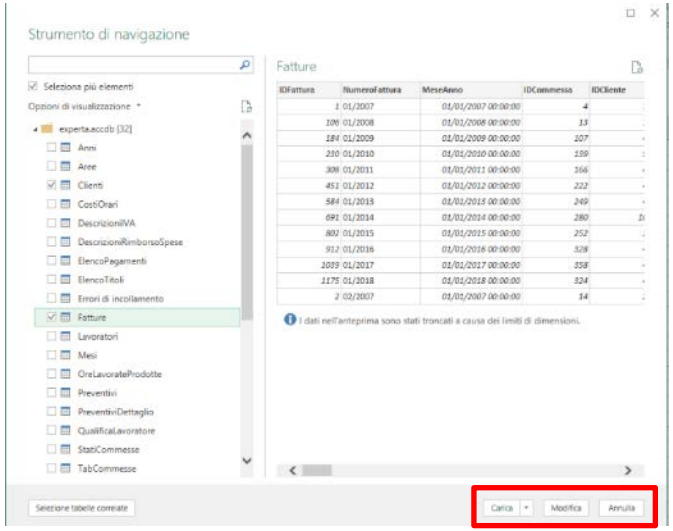
POWER QUERY: LE FONTI DATI

CONNETTERSI AD UN DATABASE

Dati > Recupera e trasforma dati > Recupera Dati > Da Database > ...

- Power Query può connettersi direttamente ad un database (SQL, Access, etc.).
- Una volta selezionata la base dati occorrerà selezionare le tabelle che si intende importare.
  - Selezionando **Carica**: le tabelle selezionate verranno caricate direttamente nella cartella di lavoro
  - Selezionando **Modifica**: si avvierà l'Editor di Power Query con cui si potranno apportare modifiche ai dati

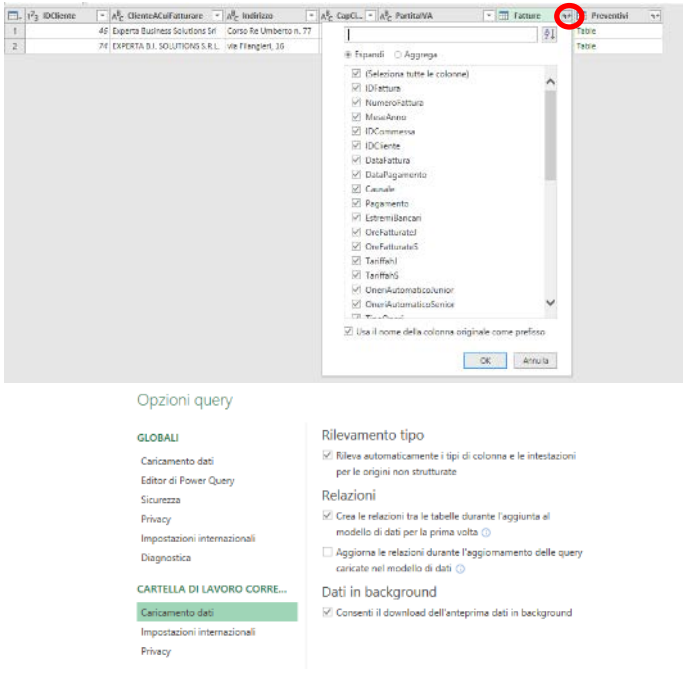




CONNETTERSI AD UN DATABASE

- Se sono presenti dei campi in comune tra le diverse tabelle Power Query ricostruisce le relazioni (o comunque le eredita dal database di origine)
- Alle tabelle che presentano delle relazioni sono aggiunte delle nuove colonne con il nome delle tabelle collegate.

	IDCliente	ClienteACuiFatturare	Indirizzo	CapCi...	PartitaIVA	Fatture	Preventivi
1	46	Experta Business Solutions Srl	Corso Re Umberto n. 77	10128 - Torino	09489580010	Table	Table
2	74	EXPERTA B.I. SOLUTIONS S.R.L.	via Filangieri, 16	10128 - Torino	10513290014	Table	Table



E' possibile navigare il dato in due modi:

- Cliccando sulla Scritta «Table» contenuta nella cella (porterà alla tabella di destinazione filtrando per il record della tabella di origine)
- Cliccando sul simbolo con due frecce nell'intestazione della colonna e selezionando i dati che si vogliono inserire (Espansione tabella)

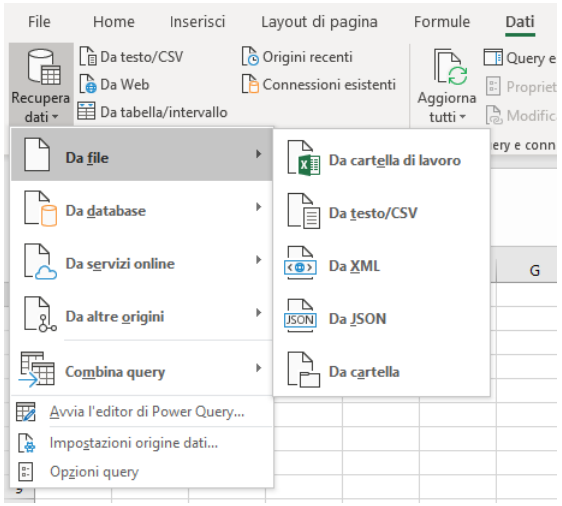
E' possibile disattivare la rilevazione delle relazioni (**Editor di Power Query > File > Opzioni e Impostazioni > Opzioni Query > Caricamento dati > Relazioni**).

Oppure, semplicemente chiedere di non visualizzare le colonne (**primo passaggio query > simbolo ingranaggio > Avanzate**)

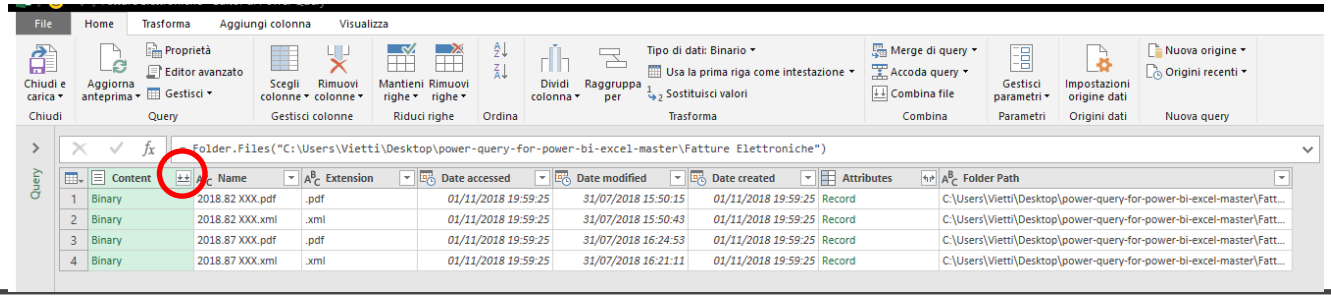
CONNETTERSI AD UN FILE

Dati > Recupera e trasforma dati > Recupera Dati > Da File > …

- Power Query può connettersi numerosi tipi di file:
  - **Altre cartelle di lavoro Excel**: una volta selezionato il file (come per i database occorrerà selezionare cosa importare). Comparirà un elenco dei fogli di lavoro e delle tabelle. E' consigliabile impostare i dati da importare in tabelle – **Inserisci > Tabelle > Tabella (CTRL+T).**
  - **File di Testo e .CSV**
  - **File XML**: questa tipologia di file non ha una struttura tabulare ma solitamente presenta delle gerarchie ad albero, la navigazione avviene esattamente nello stesso modo descritto nelle pagina precedenza per le relazioni tra database).
  - **File JSON**



- **Da cartella**: è possibile selezionare una cartella in cui sono stati salvati più file e importarli tutti insieme (accodandoli). Questa funzionalità funziona bene se i file condividono la stessa struttura. Per Combinare i file in un'unica tabella cliccare sul simbolo della doppia freccia nell'intestazione della prima colonna «Content»



CONNETTERSI AD UN FILE

Dati > Recupera e trasforma dati > Recupera Dati > Da File > …

Da PDF: da mese di agosto 2020 per gli utenti Microsoft 365 è stata attivata la possibilità di importare dati da file PDF. La funzionalità riconosce le Tabelle presenti sul file (anche se suddivise in più pagine).

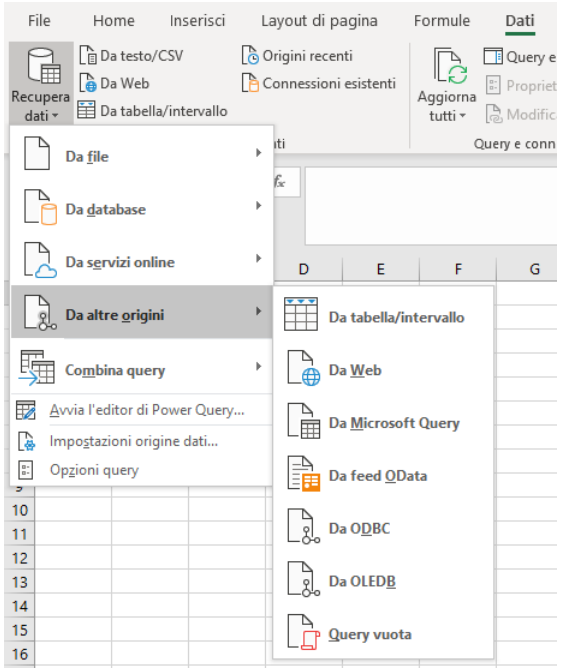
Guarda il video tutorial di questa funzionalità su YouTube:
EXCEL - POWER QUERY: Importare dati da file PDF (link)

Guarda il video tutorial di questa funzionalità su YouTube:
EXCEL - POWER QUERY: Importare da PDF su WEB - I codici tributo dal sito dell'Agenzia delle Entrate (link)

CONNETTERSI AD ALTRE ORIGINI

Dati > Recupera e trasforma dati > Recupera Dati > Da altre origini > …

- Power Query può connettersi a numerose altre origini. In particolare:
  - **Web**: sarà sufficiente indicare l'indirizzo web della pagina contenente la tabella che si vuole importare
  - **Da Tabella / Intervallo**: ossia da dati presenti all'interno della cartella di lavoro in cui siamo. In questo caso i dati devo essere contenuti necessariamente in una tabella - **Inserisci > Tabelle > Tabella (CTRL+T)**. Se così non è prima di attivare la procedura di importazione Power Query / Excel chiederà se si vuole convertire l'intervallo in tabella (ma è consigliabile creare la tabella manualmente – in modo da potergli dare un Nome «parlante»).
  - **Query vuota**: crea una query che non acquisisce dati da fonti esterne… può essere utile per creare manualmente degli elenchi attraverso le funzioni di Power Query.

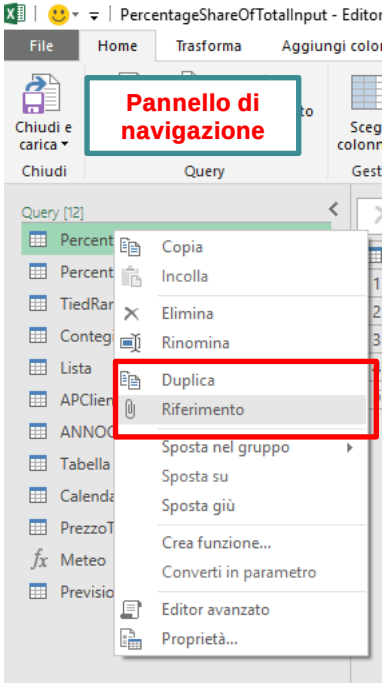


Per modificare l'origine è sufficiente cliccare sul simbolo dell'ingranaggio a sinistra del passaggio Origine (primo passaggio della query). E' possibile tuttavia creare origine esterne dinamiche con alcuni semplici passaggi

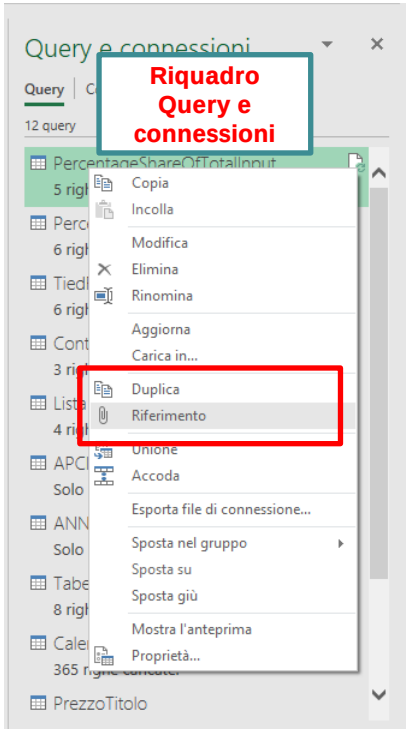
Guarda il video tutorial di questa funzionalità su YouTube:
EXCEL - POWER QUERY: Origine esterna dinamica (aggiornata in automatico) (link)

CONNETTERSI AD UNA QUERY

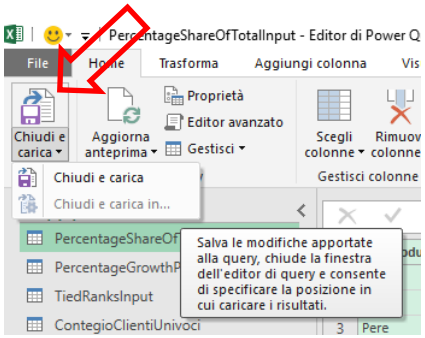
- Power Query può creare una query che ha come origine il risultato di un'altra query: tecnicamente si tratta di un «Riferimento». Può sembrare strano ma in alcuni casi suddividere una query molto estesa in piccole parti può risultare utile per i successivi sviluppi (ad es: se occorre fare due elaborazioni sugli stessi dati già in parte ripuliti).



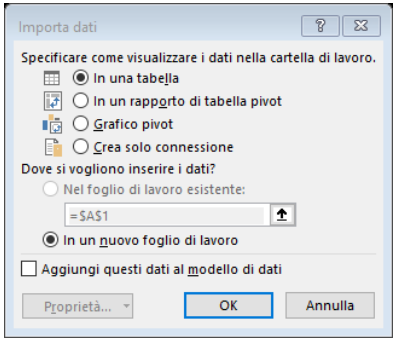
- Per creare una query da un'altra è sufficiente selezionare con il pulsante dx del mouse la query di origine (dal Riquadro Query e connessioni in Excel, oppure dal pannello di navigazione dell'Editor di Power Query) e selezionare **Riferimento.** Verrà creata una query con un solo passaggio «Origine» dato dal risultato della query selezionata
- E' anche possibile fare una copia di una query di origine per poi modificarla successivamente (in questo caso vengono copiati tutti i singoli passaggi ed eventuali modifiche sulla query di origine non influiscono sulla copia). Selezionare **Duplica.**



CHIUDI E CARICA...

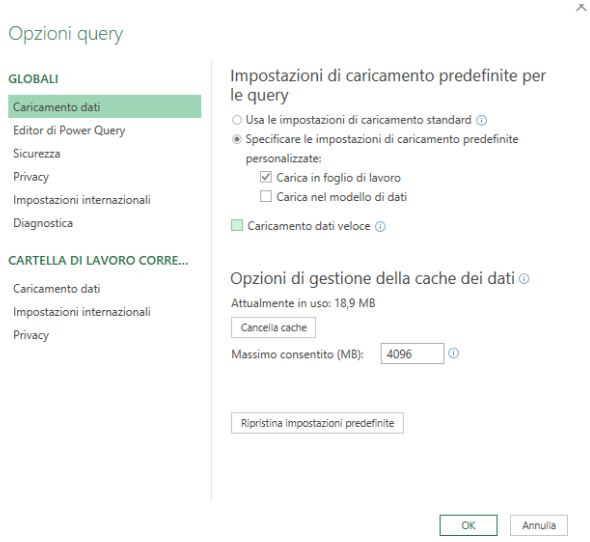


- Quando si è soddisfatti della propria query è possibile uscire dall'Editor di Power Query cliccando sul comando **Chiudi e carica**
- La prima volta sarà attivo il comando **Chiudi e carica in...** che permette di scegliere dove caricare i dati (cliccando su Chiudi e carica verranno applicati i parametri di default in caso di primo caricamento, o precedentemente impostati in caso di caricamenti successivi).
- I parametri di default sono definiti in **Editor di Power Query > File > Opzioni e impostazioni > Opzioni Query > Caricamento dati**



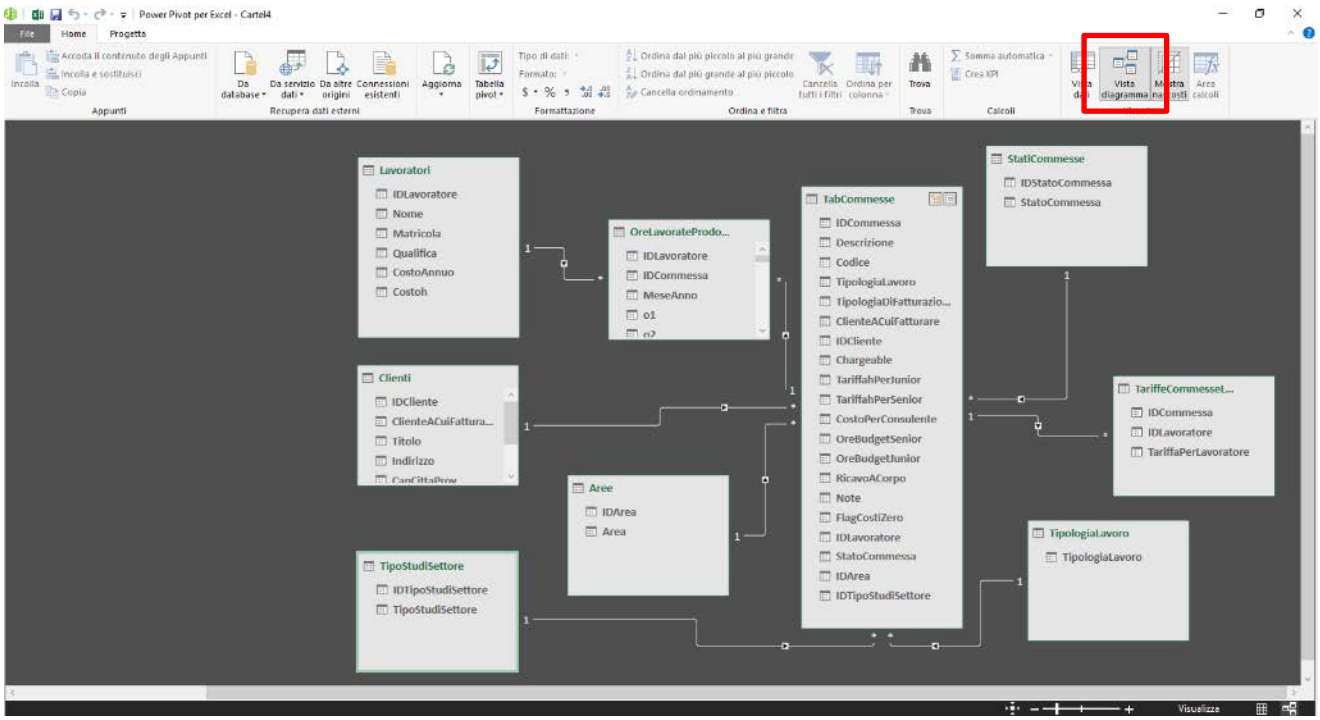
- Se si sceglie Chiudi e Carica in... si aprirà una finestra di dialogo in cui si potrà scegliere se:
  - Caricare i dati nella cartella di lavoro (tabella / Tabella Pivot / Grafico Pivot)
  - Creare solo una connessione (i dati non saranno visibili sulla cartella)
  - Aggiungere i dati al modello di dati

Queste impostazioni possono essere modificate selezionando il comando **Carica in...** dal menù che compare cliccando sulla query (dx mouse) nel Riquadro Query e connessioni.



VISUALIZZARE LE RELAZIONI

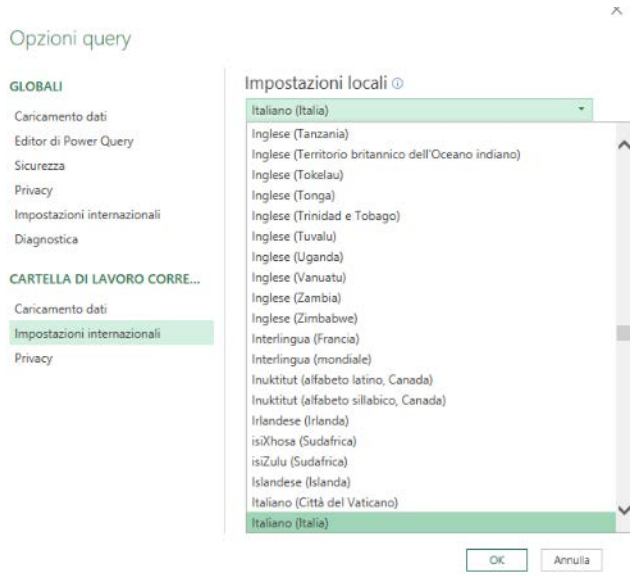
- Se i dati vengono aggiunti al modello di dati, sarà possibile visualizzare graficamente le relazioni che sono state riconosciute / definite da Power Query, aprendo il modello di dati attraverso la scheda Power Pivot
- **Power Pivot > Modello di dati > Gestisci** aprirà l'Editor di Power Pivot **Home > Visualizza > Vista Diagramma**



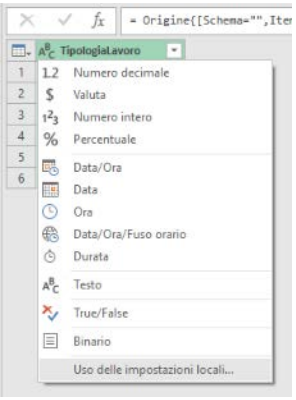
LE IMPOSTAZIONI LOCALI

- Le impostazioni locali (Nazione) sono fondamentali per importare correttamente date (GG/MM/AA oppure MM/GG/AA) e numeri (separatori delle migliaia e dei decimali).

Le impostazioni di default sono definite in: **Editor di Power Query > File > Opzioni e Impostazioni > Opzioni Query > Impostazioni internazionali**

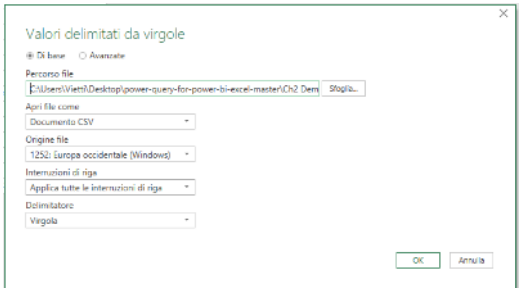


Possono tuttavia essere definite impostazioni specifiche per ciascuna colonna: **click sul simbolo del tipo di dati nell'intestazione > Uso delle impostazioni locali...**



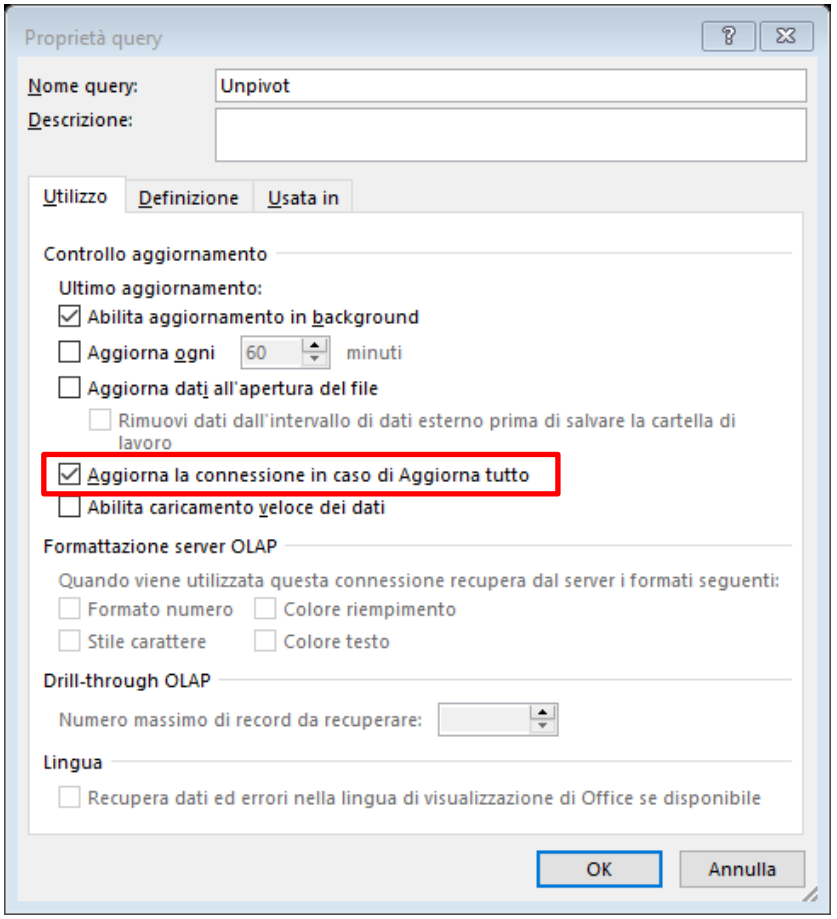
Nel caso dei file CSV e di testo le impostazioni possono essere modificate per tutto il file: **primo passaggio query «Origine» > simbolo ingranaggio**

NB: in questa finestra è anche possibile modificare il percorso del file di origine!!!



AGGIORNARE UNA QUERY

- Per aggiornare una query, sarà sufficiente selezionarla (dx mouse) nel riquadro Query e connessioni e cliccare su **Aggiorna**, oppure cliccare sul simbolo dell'aggiornamento che compare in a sinistra del nome della query.
- Il comando **Dati > Query e connessioni > Aggiorna tutti**, permette di aggiornare tutte le query contemporaneamente (a meno che non sia stata deselezionata l'opzione «Aggiorna la connessione in caso di Aggiorna tutto» nella finestra delle opzioni della singola query – di default il flag è attivo)
- Nella stessa finestra di dialogo è possibile impostare l'aggiornamento automatico della query (ogni X minuti, all'apertura del file).



POWER QUERY: TRASFORMARE I DATI

I PASSAGGI («STEPS»)

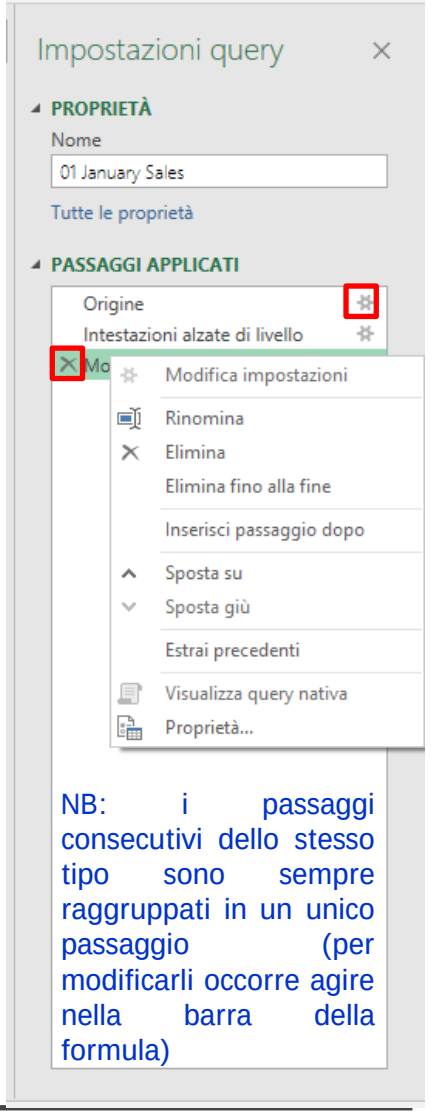
Il riquadro Impostazioni Query produce l'elenco dei passaggi applicati nella query selezionata. Cliccando su ogni passaggio è possibile visualizzare l'anteprima del risultato dello stesso.

E' possibile:

- **Modificare il passaggio**: selezionandolo e modificando manualmente il codice all'interno della barra della formula oppure cliccando sul simbolo dell'ingranaggio (che appare quando il passaggio è stato creato utilizzando l'interfaccia utente) oppure **dx mouse > Modifica impostazioni** oppure semplicemente doppio click. Nel caso del primo passaggio «Origine» questa operazione permette di modificare il percorso del file di origine, il formato e le impostazioni locali.
- **Eliminare il passaggio**: sarà sufficiente cliccare sulla X che appare a sinistra del nome, oppure **dx mouse > Elimina**. Se si vogliono eliminare tutti i passaggi successivi **dx mouse > Elimina fino alla fine**.
- **Modificare l'ordine dei passaggi**: è sufficiente trascinare il passaggio nell'ordine desiderato oppure **dx mouse > Sposta su/giù**.
- **Rinominare un passaggio**: **dx mouse > Rinomina**. E' consigliabile per i passaggi «personalizzati».
- **Inserire passaggi intermedi**: selezionando il passaggio dopo il quale si vuole inserirne uno nuovo e procedere con le modifiche oppure **dx mouse > Inserisci passaggio dopo**.
- Creare una nuova query con i passaggi precedenti a quello selezionato: **dx mouse > Estrai precedenti**.

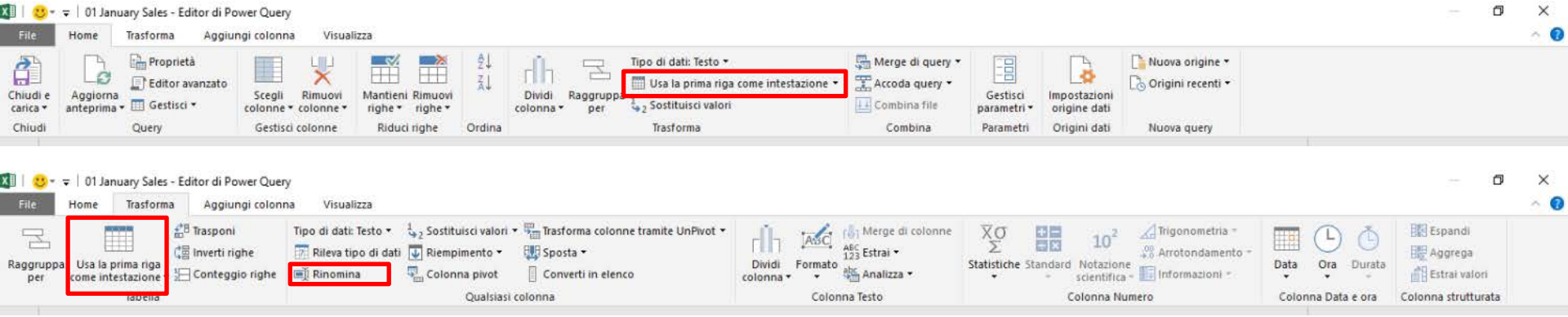
Attenzione tutte le modifiche potrebbero risultare in errori nei passaggi successivi (il programma avverte quando questo può avvenire) e non esiste la funzionalità «torna indietro – Undo» (CTRL+Z).

Power Query non ricarica i dati ad ogni modifica. Se i dati di origine sono stati modificati occorre selezionare il comando **Home > Query > Aggiorna anteprima**, per visualizzare il risultato della query con i dati aggiornati.



NB: i passaggi consecutivi dello stesso tipo sono sempre raggruppati in un unico passaggio (per modificarli occorre agire nella barra della formula)

TRASFORMARE LE COLONNE (CAMPI)

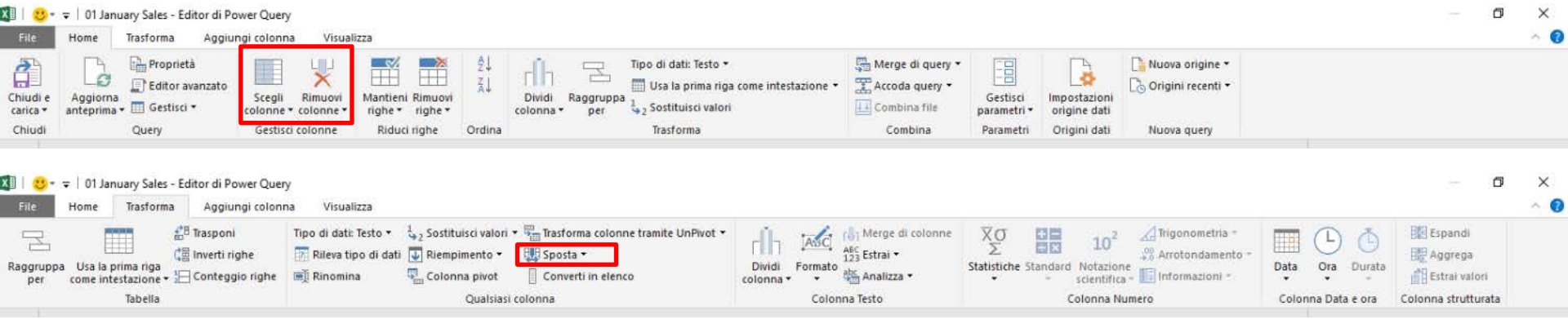


Una delle prime cose da verificare dopo aver importato i dati è che le colonne (i campi) siano nel posto giusto con il nome corretto e impostate con il Tipo dati (data, numero, testo) appropriato al contenuto. I comandi per effettuare queste operazioni sono accessibili da **Home** oppure da **Trasforma** (in alcuni casi sono duplicati) o comunque spesso con **dx mouse sull'intestazione della colonna**.

Per rinominare le colonne:

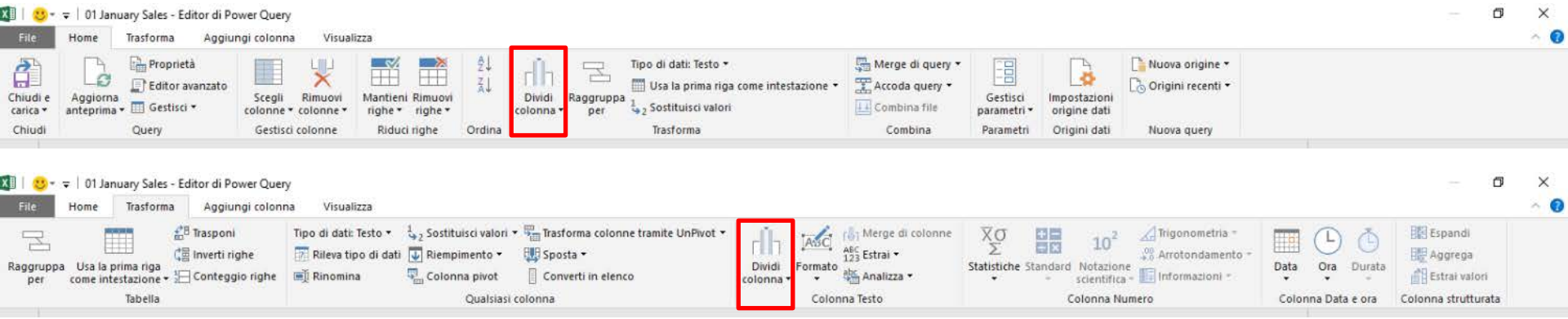
- È possibile selezionare il comando **Trasforma > Rinomina** oppure fare doppio click sull'intestazione nell'anteprima
- Il comando **Usa la prima riga come intestazione**, permette di trasformare la prima riga importata in intestazione delle colonne (spesso Power Query lo fa in automatico in fase di importazione). Nel dettaglio del comando è disponibile anche l'istruzione inversa **Usa intestazioni come prima riga**.

TRASFORMARE LE COLONNE (CAMPI)



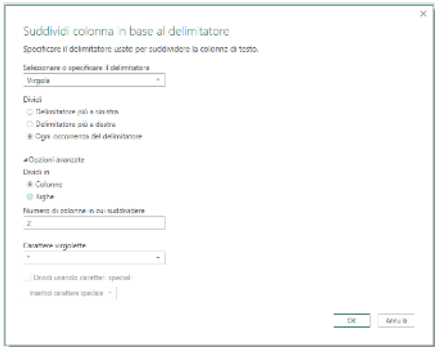
- La selezione delle colonne può avvenire direttamente cliccando l'intestazione in anteprima oppure utilizzando il comando **Home > Gestisci colonne > Scegli colonne / Vai alla colonna**. Nel primo caso la selezione multipla avviene tenendo promuto ALT (consecutive) o CTRL (non consecutive).
- Per eliminare una o più colonne è sufficiente selezionarle e premere CANC (verrà creato un passaggio di eliminazione) oppure utilizzare i comandi: **Home > Gestisci colonne > Rimuovi colonne / Rimuovi altre colonne**.
- Per modificare l'ordine delle colonne sarà sufficiente spostare con il mouse la / le colonne selezionate nell'anteprima, oppure utilizzare i comandi **Trasforma > Qualsiasi colonna > Sposta > Sinistra / Destra / All'inizio / Alla fine**.

TRASFORMARE LE COLONNE (CAMPI)



- I dati contenuti in una colonna possono essere suddivisi in più campi: **Home > Trasforma > …** oppure **Trasforma > Colonna Testo >…**

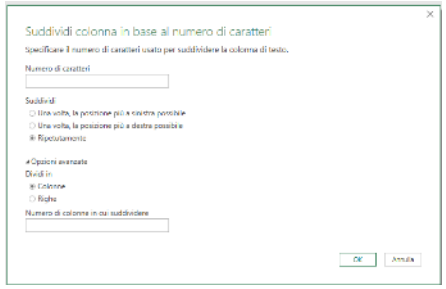
… Dividi colonna > in base ad un delimitatore: occorrerà dichiarare il delimitatore



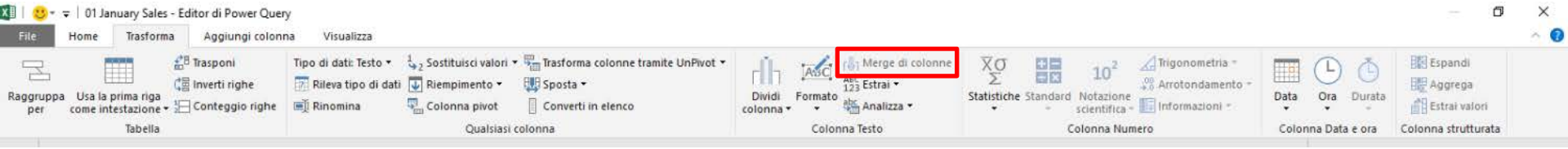
Nelle Opzioni avanzate sarà possibile:

- Dichiarare il numero di colonne (l'eventuale testo in eccesso verrà troncato)
- Scegliere se dividere i dati in colonne (default) o in righe (suddividerà i dati nella stessa colonna creando tante righe).

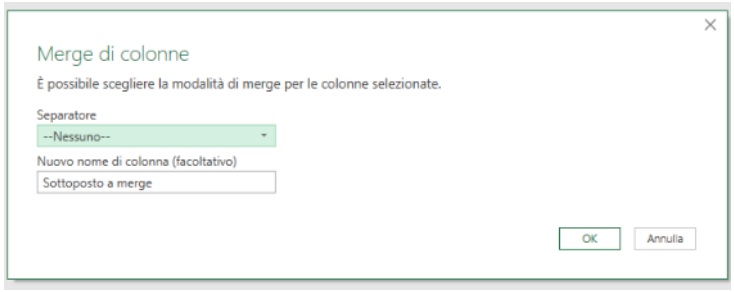
… Dividi colonna > in base al numero di caratteri: occorrerà dichiarare il delimitatore



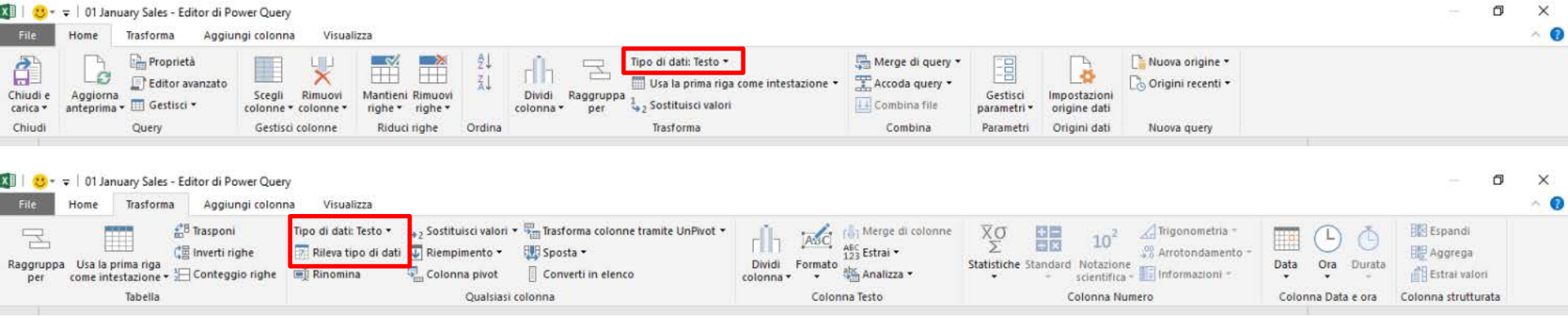
TRASFORMARE LE COLONNE (CAMPI)



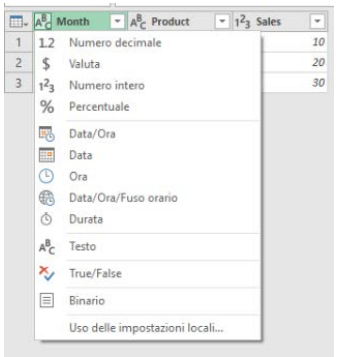
- I dati contenuti in più colonne selezionate possono essere raggruppati in un unico campo: **Trasforma > Colonna Testo > Merge di colonne**
  - Verrà richiesto quale separatore applicare e il nome della nuova colonna
  - L'ordine con cui le colonne vengono selezionate è lo stesso ordine con cui verranno uniti i dati
  - Il comando creerà una nuova colonna ed eliminerà quelle di origine
- Se lo stesso comando viene selezionato da : **Aggiungi > Da Testo > Merge di colonne**, la colonna verrà aggiunta e quelle originali verranno mantenute.



TRASFORMARE LE COLONNE (CAMPI)



- Per definire il tipo dati di una colonna (è importante che questo sia corretto per poter utilizzare a pieno le funzionalità di Power Query) occorre selezionare la colonna e attivare il comando **Home > Trasforma > Tipo di dati** oppure **Trasforma > Qualsiasi colonna > Tipo di dati** oppure **cliccare sul simbolo del Tipo dati** a sinistra nell'intestazione della colonna.
- Power Query in fase di importazione e quando percepisce dei cambiamenti nel tipo dati attiva automaticamente il comando **Trasforma > Qualsiasi colonna > Rileva tipo dati** generando un passaggio (che può essere eliminato). Il comando può essere richiamato manualmente.
- Le tipologie sono sufficientemente intuitive, alcuni chiarimenti sono tuttavia necessari:
  - Tipo dati **Binario**: serve per salvare sequenze di byte ed è utilizzato per colonne che contengono riferimenti a file del disco
  - Tipo dati **Durata**: serve per i dati che rappresentano un lasso di tempo tra due date, date/ora, date/ora/fuso
  - Tipo dati **Vero/Falso**: serve per le colonne che contengono valori logici (es: verifica di un test)
  - Tipo dati **Qualsiasi**: non è possibile selezionarlo ma appare in automatico quando Power Query non riesce ad attribuire una tipologia specifica (e sempre meglio che non ce ne siano)

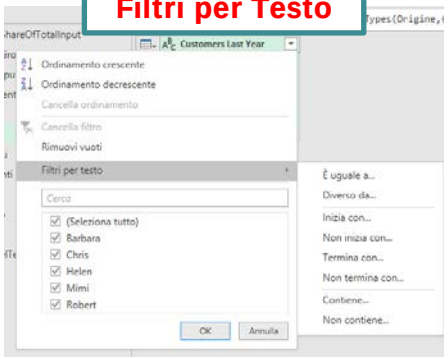


FILTRARE LE RIGHE (RECORD)

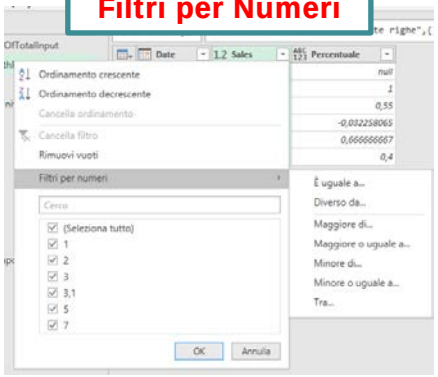
- Le righe possono essere filtrate attraverso l'utilizzo del **Filtro Automatico** (del tutto analogo a quello di Excel), si può operare sull'elenco selezionando i valori da escludere (eventualmente utilizzando la barra di ricerca). La finestra del filtro presenterà le prime 1.000 occorrenze distinte (scorrendo al fondo si potrà scegliere di aggiungerne altre 1.000, e così via). A seconda del tipo di dati il filtro presenterà delle opzioni personalizzate:

Filtri per Date

Filtri per Testo



Filtri per Numeri



- L'inserimento di un filtro comporterà la creazione di un passaggio (per eliminare o modificare il filtro occorrerà agire sul passaggio).
- A differenza dei filtri in Excel, in Power Pivot (essendo un passaggio delle query) i filtri vengo riapplicati in fase di aggiornamento dei dati.

FILTRARE LE RIGHE (RECORD)

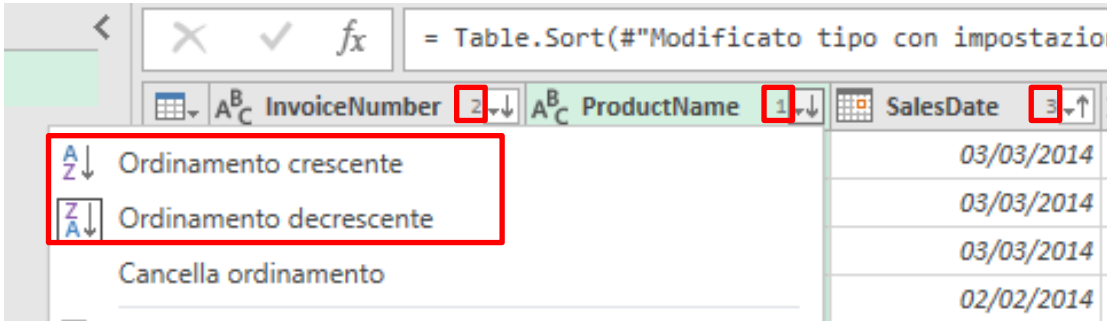
- Le righe possono essere filtrate attraverso il comando **Home > Riduci Righe > Mantieni / Rimuovi Righe…** che permette di mantenere o rimuovere:
  - **…Prime / Ultime / Intervallo di righe**: si aprirà una finestra di dialogo con la richiesta di specificare i parametri
  - **…Righe con Errori**: di default sono mantenute ma possono essere rimosse (se si seleziona una colonna verranno eliminate tutti i record con presentano un errore in quella colonna, senza selezione si applica a tutta la tabella)
  - **…Duplicati**: si applica ai duplicati delle colonne selezionate
  - **…Righe vuote**: si applica alla selezione (attenzione il valore *null*, non è vuoto)
- I comandi sono anche accessibili dal simbolo della tabella in alto a sinistra nel riquadro di anteprima.

FILTRARE LE RIGHE (RECORD)

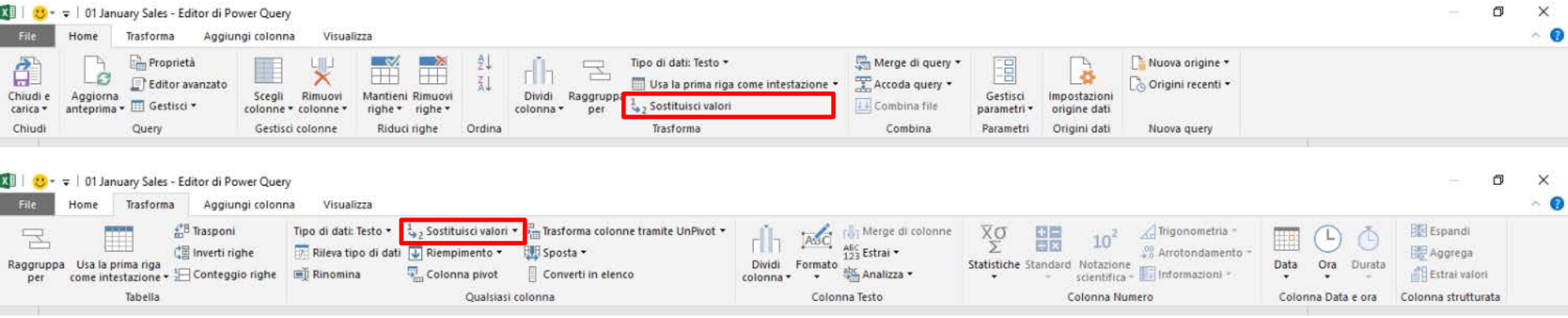
- Il comando **Home > Riduci Righe > Rimuovi Righe > Rimuovi righe alternate** permette di rimuovere righe secondo una regola ripetuta. La finestra di dialogo che si apre chiede di inserire i seguenti parametri
  - **Prima riga da rimuovere**: indicare il numero della prima riga da rimuovere
  - **Numero di righe da rimuovere**: numero di righe da rimuovere comprendendo la prima
  - **Numero di righe da mantenere**: numero di righe da mantenere dopo l'intervallo rimosso
- La procedura continuerò con lo stesso schema rimuovi/mantieni fino alla fine delle righe della tabella.

ORDINARE LE RIGHE (RECORD)

- Le righe di una query possono essere ordinate attraverso i comandi **Ordinamento crescente / Ordinamento decrescente** accessibili dal filtro automatico di ciascuna colonna oppure da **Home > Ordina > Ordinamento crescente / Ordinamento decrescente**.
- Power query inserirà una piccola freccia (dall'alto in basso o viceversa) nel pulsantino del filtro automatico della colonna.
- Power query memorizza l'ordine con cui le colonne vengono ordinate, inserendo un numero progressivo nell'intestazione della colonna. Tutti gli ordinamenti consecutivi sono salvati in un unico passaggio. Eventuali passaggi successivi di ordinamento sovrascriveranno gli ordinamenti precedenti.



MODIFICARE I VALORI



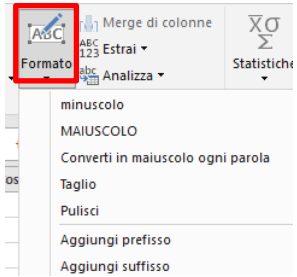
- Il comando **Sostituisci valori** accessibile sia da **Home > Trasforma** sia da **Trasforma > Qualsiasi colonna** permette di sostituire specifiche occorrenze.
- Si aprirà una finestra di dialogo che richiederà il valore da trovare e quello con cui sostituirlo.

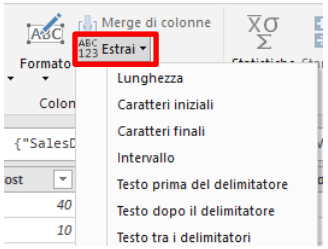
ATTENZIONE: Power Query è case sensitive! → «Arancia» è diverso da «arancia»

- Per **cancellare un valore** è sufficiente sostituirlo lasciando il campo Sostituisci con vuoto
- Per **sostituire valori vuoti con *null*** sarà sufficiente lasciare il campo valore da trovare vuoto e sostituisci con *null*
- Per **sostituire altri valori con *null*** sarà sufficiente procedere nello stesso modo (assicurandosi che l'opzione avanzata «Confronta intero contenuto della cella» sia selezionata)
- Per evitare errori di digitazione nel campo valore da trovare potrebbe essere utile attivare il comando con dx mouse sul valore che si vuole sostituire

MODIFICARE I VALORI (TESTO)

- In **Trasforma > Colonna Testo > …** sono accessibili alcuni comandi utili a modificare il testo contenuto in una colonna:
  - **…Formato > …** permette modificare **minuscolo / MAIUSCOLO / Iniziali Maiuscole**, il comando **Taglia** permette di eliminare eventuali spazi bianchi all'inizio e alla fine del testo, **Pulisci** consente di eliminare eventuali caratteri non stampabili
  - **…Formato > Aggiungi prefisso / suffisso**: consente di inserire una stringa all'inizio o alla fine del contenuto della cella
  - **… Estrai > Lunghezza**: sostituisce i valori con il numero di caratteri
  - **… Estrai > Caratteri iniziali / finali / Intervallo**: sostituisce i valori un numero definito dall'utente di caratteri della stringa
  - **… Estrai > Testo prima / dopo / tra delimitatori**: permette di estrarre testo se nella colonna i valori sono separati da delimitatori





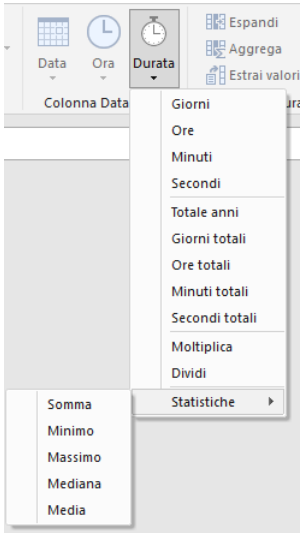
NB: Gli stessi comandi sono presenti in **Aggiungi colonna > Da testo > …** in questo caso il comando non agisce sulla colonna selezionata ma ne crea una nuova

MODIFICARE I VALORI (NUMERI)

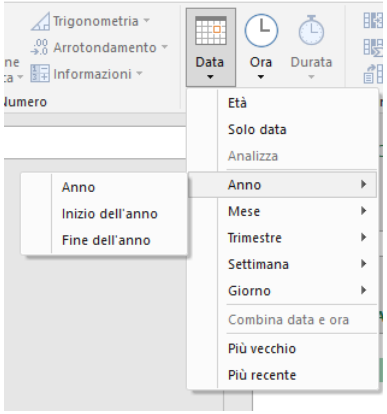
- In **Trasforma > Colonna Numero > …** sono accessibili alcuni comandi utili a modificare i valori numerici contenuto in una colonna o fare dei calcoli sul totale della colonna:
  - **…Statistiche > …** consente di ottenere informazioni statistiche sui valori inseriti in una colonna (Somma, Media, Massimo, Minimo, Mediana, Dev Std, Conteggio valori e Conteggio valori distinti). Queste informazioni potranno essere poi utilizzate per costruire formule più complesse.
  - **… Standard > …** permette di effettuare calcoli il cui risultato sostituirà i valori contenuti nella colonna (somme, sottrazioni, moltiplicazioni, divisioni, divisione intera, modulo, percentuale) per costanti o valori contenuti in altre colonne.
  - **… Notazione scientifica > …** consente di calcolare, tra le altre cose, potenza e valore assoluto
  - **… Trigonometria > …** permette di effettuare calcoli trigonometrici
  - **… Arrotondamento > …** consente di arrotondare per effetto o per difetto all'unità oppure effettuare l'arrotondamento matematico tradizionale ad un determinato decimale (valori negativi permette di arrotondare per decine, centinaia, migliaia,…)
  - **… Informazioni > …** restituisce se il valore è positivo / negativo (attraverso valori logici VERO/FALSO) oppure il segno (1 se positivo -1 se negativo 0 se non è un numero)

NB: Gli stessi comandi sono presenti in **Aggiungi colonna > Da Numero > …** in questo caso il comando non agisce sulla colonna selezionata ma ne crea una nuova

MODIFICARE I VALORI (DATE e ORE)



- In **Trasforma > Colonna Data e ora > …** sono accessibili alcuni comandi utili a modificare o elaborare i dati delle colonne contenenti informazioni su data, ora o durata:
  - **…Data / Ora > …** permette estrapolare anno/ mese/ trimestre/ settimana/ giorno / ora / minuto / secondo (effettuando anche delle elaborazioni, es: inizio, fine anno o mese… oppure convertire il nome del mese / giorno)
  - **…Data / Ora > Più vecchio / Più recente**: estrae il dato più vecchio o più recente dall'elenco. Queste informazioni potranno essere poi utilizzate per costruire formule più complesse.



- **…Data / Ora > Età**: restituisce la durata tra le date contenute nella colonna e la data del pc in uso
- **…Durata > …**: permette di estrarre dati arrotondati da un campo contenenti durate o effettuare calcoli da utilizzare per formule più complesse.

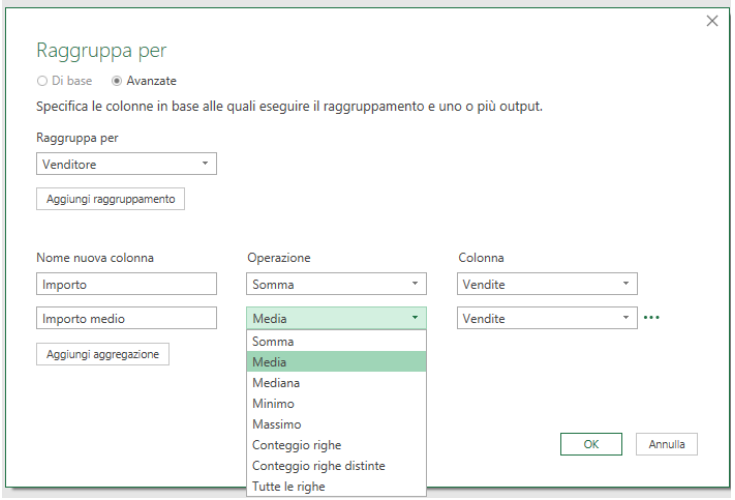
NB: Gli stessi comandi sono presenti in **Aggiungi colonna > Da Data e Ora > …** in questo caso il comando non agisce sulla colonna selezionata ma ne crea una nuova

MODIFICARE I VALORI – IL COMANDO RIEMPIMENTO

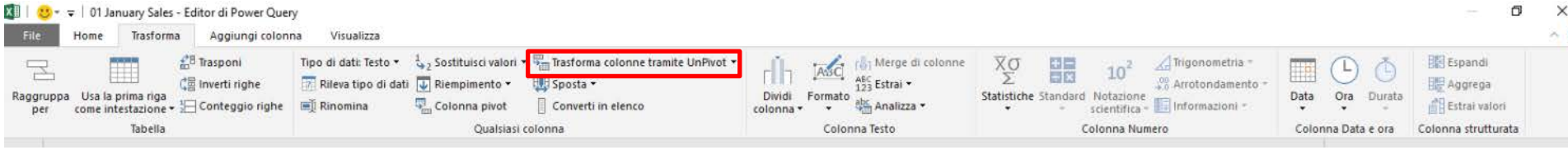
- Il comando **Trasforma > Qualsiasi colonna > Riempimento > ...** permette di sostituire i valori *null* di una colonna riportando il primo valore non nullo presente nei record sopra (**...In basso**) o sotto (**...In alto**).
- Attenzione il comando non analizza le altre colonne, ma semplicemente sostituisce i valori *null*, quindi occorre accertarsi che i record sia ordinati correttamente (eventualmente impostando l'ordinamento prima di procedere).
- Attenzione il comando sostituisce i valori *null*, non i vuoti, eventualmente occorrerà procedere ad una sostituzione attraverso il comando **Trasforma > Qualsiasi colonna > Sostituisci valori**

MODIFICARE I VALORI – IL COMANDO RAGGRUPPA PER…

- Il comando **Raggruppa per** consente di raggruppare i valori sulla base del contenuto di una o più colonne effettuando uno o più dei seguenti calcoli:
  - Somma
  - Media
  - Minimo / Massimo
  - Conteggio Righe
  - Conteggio righe uniche
  - Tutte le righe (restituisce il valore tipo Tabella in una singola cella – di scarso utilizzo a meno che si intenda utilizzarlo in un codice più complesso).



MODIFICARE UNA TABELLA – UNPIVOT



- Il comando **Trasforma > Qualsiasi colonna > Trasforma colonna tramite Unpivot** consente di trasformare una tabella a doppia entrata in una basedati «piatta» riportando le intestazioni delle colonne «unpivottate» in una nuova colonna.
- Il comando può essere applicato anche solo ad alcune colonne (quelle selezionate o quelle non selezionate).

	Prod...	2010	2011	2012	2013	2014
1	Apples	5	6	7	2	3
2	Pears	1	1	4	7	8
3	Grapes	9	6	4	5	6

Unpivot

MODIFICARE UNA TABELLA – TRASPORRE

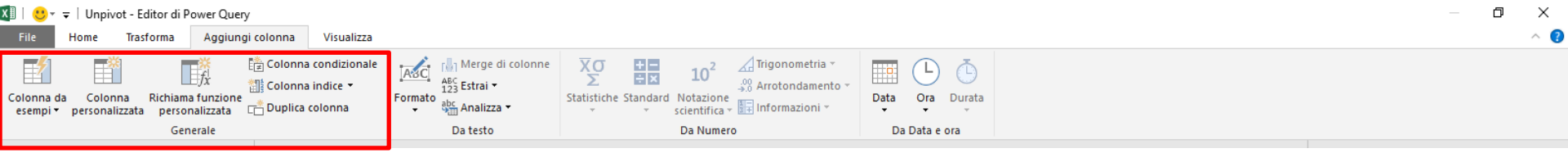
- Il comando **Trasforma > Tabella > Trasponi** consente di invertire righe e colonne di una tabella (occorre tuttavia «abbassare» le intestazioni delle colonne della tabella di origine prima di procedere alla trasformazione, in caso contrario verranno perse).

	Prod...	2010	2011	2012	2013	2014
1	Apples	5	6	7	2	3
2	Pears	1	1	4	7	8
3	Grapes	9	6	4	5	6

Trasponi

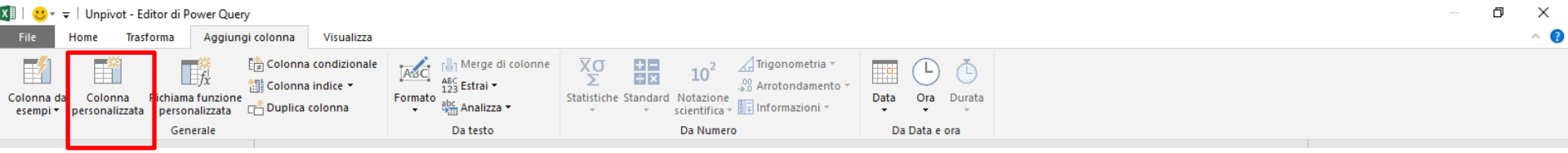
	Product	Apples	Pears	Grapes
1	2010	5	1	9
2	2011	6	1	6
3	2012	7	4	4
4	2013	2	7	5
5	2014	3	8	6

MODIFICARE UNA TABELLA – AGGIUNGERE COLONNE

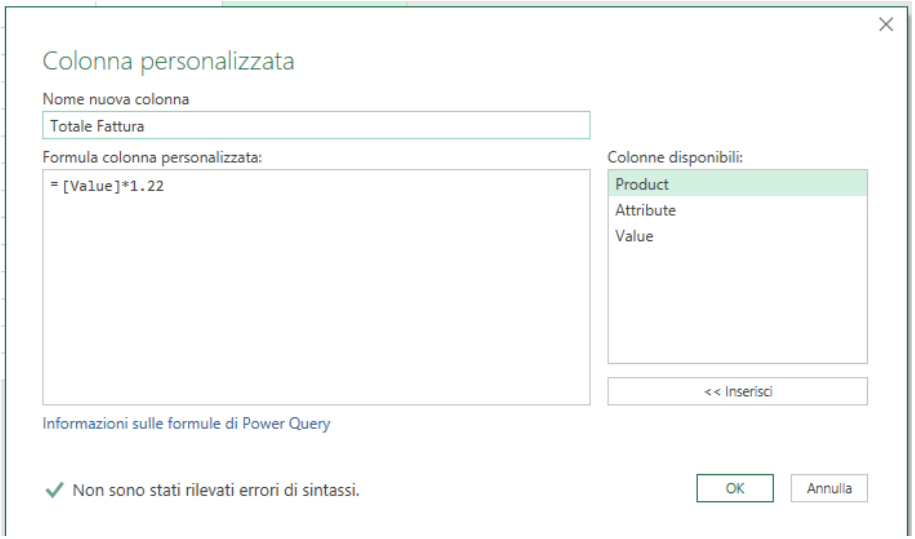


- La scheda **Aggiungi colonna > …** consente per l'appunto di inserire nuove colonne alla tabella (i gruppi Da testo / Da Numero / Da Data e ora funzionano analogamente alle opzioni della scheda Trasforma solo che al posto di sostituire i dati li inseriscono in una nuova colonna).
- **… Generale > Colonna indice > …**: inserisce una colonna con numerazione crescente delle righe partendo da 0 oppure 1 oppure da un valore personalizzato (si può personalizzare anche l'incremento)
- **… Generale > Duplica colonna**: inserisce un duplicato di una colonna che non sarà suscettibile di eventuali modifiche ai dati della colonna di origine effettuati successivamente
- **… Generale > Colonna condizionale**: inserisce una colonna con valori dichiarati a seconda del verificarsi di uno o più test sui valori corrispondenti in altre colonne (funziona analogamente alla funzione SE in Excel, con più livelli di nidificazione)
- **… Generale > Colonna da esempi > …**: funziona analogamente all'Anteprima suggerimenti di Excel, ossia permette di inserire manualmente dei valori cercando di ricostruire la regola desiderata (può essere un aiuto per ottenere il risultato desiderato senza modificare il linguaggio M nella barra della formula).

MODIFICARE UNA TABELLA – AGGIUNGERE COLONNE



- Il comando **Aggiungi colonna > Generale > Colonna personalizzata** consente di inserire colonne con calcoli personalizzati (nel caso le opzioni disponibili non siano sufficienti a soddisfare le esigenze dell'utente).
- Calcoli semplici potranno essere effettuati richiamando i nomi delle altre colonne. Formule più complesse potranno essere inserite utilizzando il linguaggio M



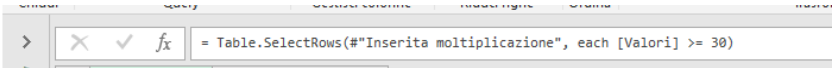
POWER QUERY: INTRODUZIONE AL LINGUAGGIO M

COSA E' IL LINGUAGGIO M?

- Per utilizzare il pieno potenziale di Power Query, occorre conoscere il linguaggio con cui vengono scritte le istruzioni delle singole query
- Il linguaggio M è il nome non ufficiale del linguaggio di programmazione utilizzato da Power Query per scrivere le formule
- Date le numerose funzionalità a cui si può accedere attraverso l'interfaccia utente e possibile utilizzare Power Query senza conoscere M, tuttavia conoscerne le basi permette di leggere (ed eventualmente modificare) le istruzioni generate automaticamente ed eventualmente scrivere autonomamente i passaggi della query.

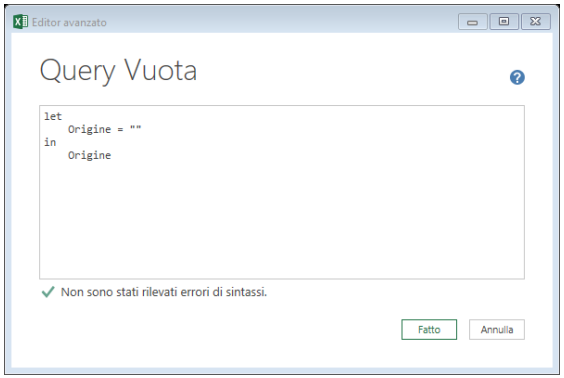
DOVE SI SCRIVE IL CODICE?

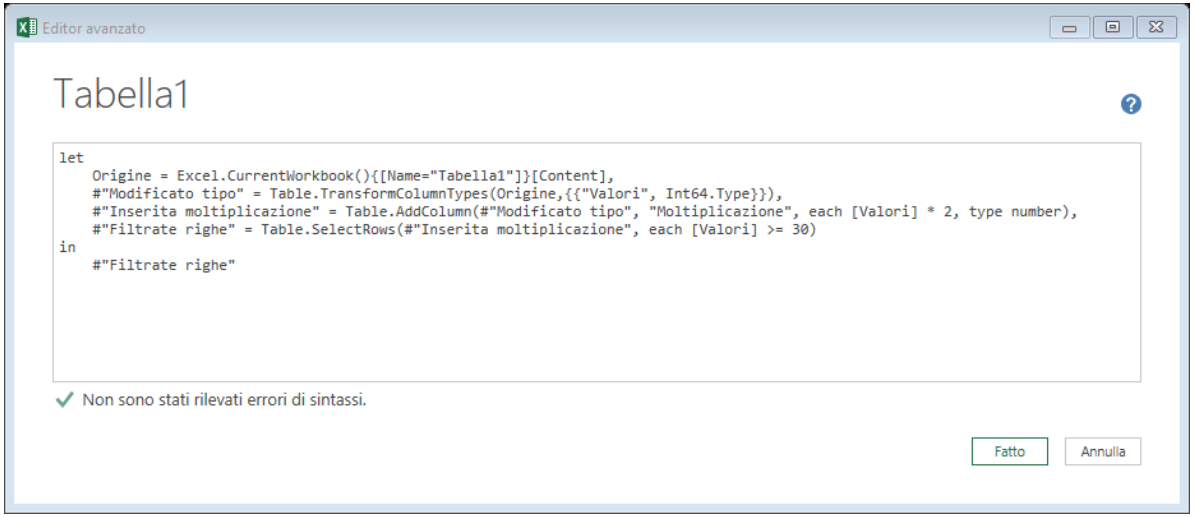
- Il codice può essere scritto:
  - Nella **barra della formula** (che permette di visualizzare il codice generato automaticamente dal programma per il passaggio selezionato)



  - Nell'**editor avanzato** (**Home > Query > Editor Avanzato**) che riporta il codice di tutta la query. L'editor, al momento, è un semplice editor di testo (ad es: non evidenzia gli errori, semplicemente presenta un indicazione se sono o meno presenti errori di sintassi)

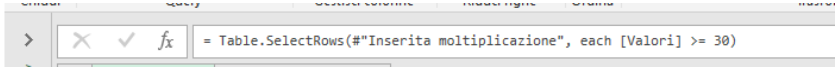
Se in fase di apertura di Power Query si seleziona Query Vuota, verrà creata una query senza codice.





M: CONCETTI BASE

- I concetti base del linguaggio M sono:
  - I **Valori**: possono essere numeri, testo o oggetti più complessi come le tabelle. Il risultato di tutto il codice M inserito in una query è sempre un valore che viene restituito dall'elaborazione (spesso e volentieri una tabella).
  - Le **Espressioni**: sono insieme di formule e funzioni che caratterizzano ogni passaggio della query (per intenderci quello che viene riportato nella barra della formula è un'espressione)



Ciò che non è così scontato è che ogni query corrisponde ad un'unica espressione in M. Questo avviene per mezzo del **Let statement** che consente ad una singola espressione di essere «spezzata» in piccole espressioni (solitamente ogni espressione – che corrisponde al passaggio – richiama il risultato di quella precedente).

```
let
    Origine = Excel.CurrentWorkbook(){[Name="Tabella1"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Valori", Int64.Type}}),
    #"Inserita moltiplicazione" = Table.AddColumn(#"Modificato tipo", "Moltiplicazione", each [Valori] * 2, type number),
    #"Filtrate righe" = Table.SelectRows(#"Inserita moltiplicazione", each [Valori] >= 30)
in
    #"Filtrate righe"
```

Il risultato della query è quello definito dopo «in» e può essere una qualsiasi delle variabili definite, ma solitamente è l'ultima – è consigliabile mantenere un ordine che permetta la lettura e la comprensione del codice. In ogni caso l'ordine non è importante.

M: CONCETTI BASE

- Il linguaggio M è CASE SENSITIVE il che significa, ad esempio, che la funzione *Table.AddColumn* non può essere digitata *Table.Addcolumn*, in questo secondo caso la query andrà in errore.

- I Tipo valori sono fondamentali nel linguaggio M. Il testo deve essere riportato tra virgolette in caso contrario la query andrà in errore. Ad esempio:

  L'espressione «Il numero» & «1» restituirà «Il numero 1»
  L'espressione «Il numero» & 1 restituirà Errore
  L'espressione «Il numero» & Number.ToText(1) restituirà «Il numero 1»

  Si può verificare il Tipo valore attraverso l'operatore is:

  1 is number restituirà VERO 1 is text restituirà FALSO

- Il separatore non è ; (come in Excel) ma , (e separa le diverse espressioni ma anche gli argomenti delle funzioni)

- Per dichiarare date, ore o durata in una espressione possono essere utilizzate le seguenti funzioni intrinseche:
  - #date(anno, mese, giorno)
  - #datetime(anno, mese, giorno, ora, minuto, secondo)
  - #datetimezone(anno, mese, giorno, ora, minuto, secondo, ore di differenza, minuti di differenza)
  - #duration(giorni, ore, minuti, secondi)

M: CONCETTI BASE

- E' possibile, anzi consigliabile, inserire commenti che spieghino i diversi passaggi del codice.
- Per inserire commenti utilizzare i seguenti simboli:

// se il commento è su una sola riga

/* se il commento è
su più righe */

I commenti non sono visibili nella barra della formula ma solo nell'Editor avanzato

```
/*     Questa query carica i dati dalla Tabella1, aggiorna il tipo valore in numeri interi, poi aggiunge
       una nuova colonna moltiplicando per due i valori e filtra i valori della prima colonna >=30 */
let

    //Carica i dati dalla Tabella
    Origine = Excel.CurrentWorkbook(){[Name="Tabella1"]}[Content],

    //Modifica il Tipo valori della colonna in numeri interi
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Valori", Int64.Type}}),

    //Moltiplica i valori per due
    #"Inserita moltiplicazione" = Table.AddColumn(#"Modificato tipo", "Moltiplicazione", each [Valori] * 2, type number),

    //Filtra i valori >=30 nella colonna di origine
    #"Filtrate righe" = Table.SelectRows(#"Inserita moltiplicazione", each [Valori] >= 30)

in
    //Restituisce la tabella definita nei passaggi precedenti
    #"Filtrate righe"
```

M: CONCETTI BASE

- Due espressioni utili possono essere:
  - *try ... otherwise* (prova ... altrimenti): questa espressione impedisce la produzione di errori quando si applicano alcune funzioni (simile alla funzione SE.ERRORE di Excel)

Valore	Number.FromText([Valore])	try Number.FromText([Valore]) otherwise 0
10	10	10
Pippo	Error	0

  - *if ... then ... else* (se ... allora ... se no): equivale alla funzione logica SE di Excel e la sintassi è simile

if [Valore] > 10 *then* «Più di 10» *else* «10 o meno»

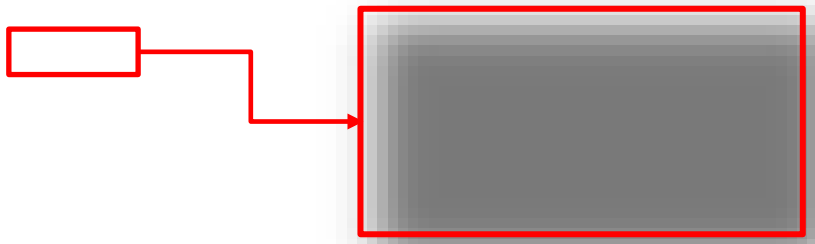
anche in Power Query è possibile nidificare le funzioni

if [Valore] > 10 *then if* [Valore] > 6 *then* «Più di 6 fino a 10 » *else* «6 o meno»

M: LISTS (ELENCHI o LISTE)

- In M i valori possono essere organizzati in **lists** (liste o elenchi), **records** e **tabelle**
- Le **liste** sono elenchi ordinati di valori (è difficile che il risultato di una query sia una lista, ma molte funzioni in M utilizzano le liste). Nel codice una lista può essere impostata manualmente inserendo i valori, separati da virgole tra parentesi graffe.

E' possibile creare liste di liste:



- Quando viene inserita una lista (manualmente o in seguito ad un passaggio della query) Power Query abilita la scheda **Strumenti per gli elenchi > Trasforma** che permette di:
  - Convertire la lista in tabella
  - Gestire gli elementi (mantieni / rimuovi)
  - Ordinare gli elementi
  - Effettuare calcoli

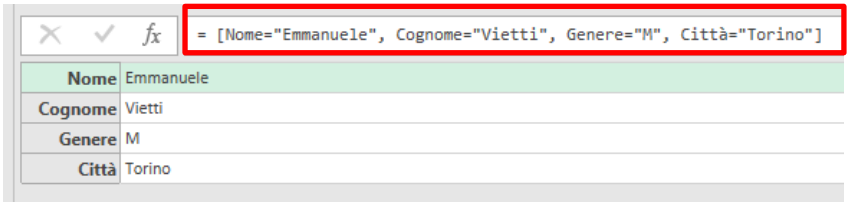
M: LISTS (ELENCHI o LISTE)

- In M i valori possono essere organizzati in **lists** (liste o elenchi), **records** e **tabelle**
- Alcune funzioni permettono di lavorare con le liste:

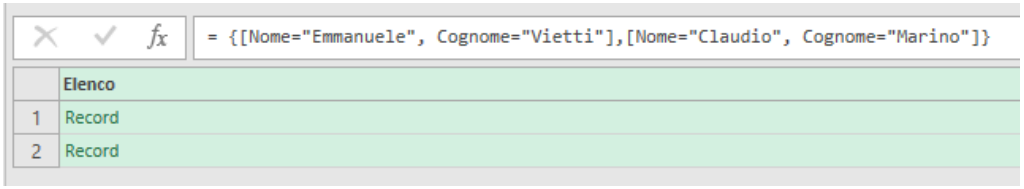
• **Generare liste**:	List.Numbers()	genera una lista di valori numerici
	List.Dates()	genera una lista di date
	Table.ToList()	converte una tabella in una lista
	Table.Column()	converte una colonna di una tabella in una lista
• **Aggregare valori**:	List.Count()	conti il numero di valori di una lista
	List.Sum()	somma i valori di una lista
	List.Average()	fa la media dei valori di una lista
	List.Min()	estrae il valore minimo di una lista
• **Ordinare i valori**:	List.Sort()	ordina una lista in ordine crescente
	List.Reverse()	inverte l'ordine
• **Filtrare i valori**:	List.First()	restituisce il primo valore
	List.FirstN()	restituisce i primi n valori
	List.Distinct()	estrae i valori univoci
	List.Select()	seleziona i valori che rispettano una condizione

M: RECORDS

- In M i valori possono essere organizzati in **lists** (liste o elenchi), **records** e **tabelle**
- I **record** possono intendersi come una tabella con una sola riga. Nel codice un record può essere impostata manualmente inserendo il nome campo=valore, separati da virgole tra parentesi quadre.



E' possibile creare liste di record:

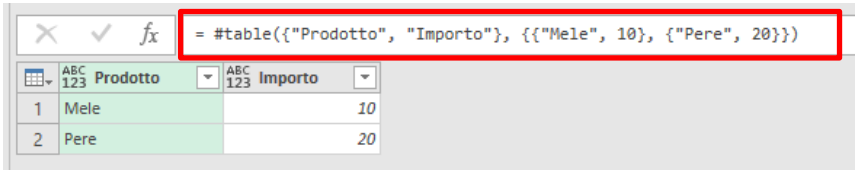


- Quando viene inserito un record (manualmente o in seguito ad un passaggio della query) Power Query abilita la scheda **Strumenti per i record > Converti** che permette di Convertire il record in tabella

M: TABELLE

- In M i valori possono essere organizzati in **lists** (liste o elenchi), **records** e **tabelle**

- Le **tabelle** sono i principali elementi strutturati in Power Query, e rappresenta nella maggior parte dei casi l'input e l'output di una query. Nel codice una tabella può essere impostata manualmente attraverso la funzione intrinseca **#table()**



E' anche possibile specificare il Tipo Valore delle colonne:
=**#table**(type table [Prodotto = text, Importo = number], {{«Mele»,10},{«Pere»,20}})

- Solitamente le tabelle non sono create manualmente ma acquisite da fonti esterne. Ad esempio la formula generata per l'acquisizione di una tabella (Vendite) dalla cartella di lavoro è:
= **Excel.CurrentWorkbook()**{[Name="Vendite"]}[Content]

- Altre funzioni che possono essere utilizzate per creare tabelle sono ad esempio:

=**Table.FromRows**({{«Mele», 10},{«Pere»,20}}, {«Prodotto, «Importo»})
=**Table.FromRecords**({[Prodotto= «Mele», Importo=10], [Prodotto= «Pere», Importo=20]})

M: RAGGRUPPARE I DATI DELLE TABELLE

TabellaVendite

Trimestre	Mese	Importo
Q1	Gennaio	2
Q1	Febbraio	3
Q1	Marzo	5
Q2	Aprile	3
Q2	Maggio	6
Q2	Giugno	8
Q3	Luglio	7
Q3	Agosto	6
Q3	Settembre	6
Q4	Ottobre	9
Q4	Novembre	8
Q4	Dicembre	5

```
let
    //Carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Conta le righe
    ConteggioRighe = Table.RowCount(Origine)

in
    ConteggioRighe
```

La funzione Table.RowCount() restituisce il conteggio delle righe della tabella di origine in forma di record.

Se si vogliono effettuare altri calcoli occorre trasformare il risultato in una tabella attraverso la funzione Table.Group()

```
let
    //Carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Conta le righe e restituisce una tabella
    TabellaConteggioRighe= Table.Group(Origine, {}, {{"Conteggio Righe", each Table.RowCount(_), type number}})

in
    TabellaConteggioRighe
```

1.2 Conteggio Righe
12

M: RAGGRUPPARE I DATI DELLE TABELLE

TabellaVendite

Trimestre	Mese	Importo
Q1	Gennaio	2
Q1	Febbraio	3
Q1	Marzo	5
Q2	Aprile	3
Q2	Maggio	6
Q2	Giugno	8
Q3	Luglio	7
Q3	Agosto	6
Q3	Settembre	6
Q4	Ottobre	9
Q4	Novembre	8
Q4	Dicembre	5

```
let
    //Carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Raggruppa per trimestre, conta le righe e somma gli importi
    RigheRaggruppate= Table.Group(
                        Origine,
                        {"Trimestre"},
                        {{"Conteggio Righe", each Table.RowCount(_), type number},
                        {"Somma di Importo", each List.Sum([Importo]), type number}}
                        )
in
    RigheRaggruppate
```

	Trimestre	Conteggio Righe	Somma di Importo
1	Q1	3	10
2	Q2	3	17
3	Q3	3	19
4	Q4	3	22

M: RAGGRUPPARE I DATI DELLE TABELLE

VenditeGiornaliere

Data	WeekdayOrWeekend	Importo	Quanti
01-gen-14	Weekday	5	2
02-gen-14	Weekday	4	1
03-gen-14	Weekday	2	1
04-gen-14	Weekend	6	1
05-gen-14	Weekend	8	5
06-gen-14	Weekday	6	3
07-gen-14	Weekday	4	2
08-gen-14	Weekday	5	1
09-gen-14	Weekday	5	2
10-gen-14	Weekday	2	3
11-gen-14	Weekend	1	1
12-gen-14	Weekend	3	1

```
let
    //Carica i dati dalla tabella VenditeGiornaliere
    Origine= Excel.CurrentWorkbook(){[Name="VenditeGiornaliere"]}[Content],

    //Conta le righe raggruppando per il campo WeekdayOrWeekend senza considerare l'ordine
    RigheRaggruppate= Table.Group(Origine,
                                  {"WeekdayOrWeekend"},
                                  {{"Conteggio Righe", each Table.RowCount(_), type number}},
                                  GroupKind.Global)

in
    RigheRaggruppate
```

	WeekdayOrWeekend	Conteggio Righe
1	Weekday	8
2	Weekend	4

```
let
    //Carica i dati dalla tabella VenditeGiornaliere
    Origine= Excel.CurrentWorkbook(){[Name="VenditeGiornaliere"]}[Content],

    //Conta le righe raggruppando per il campo WeekdayOrWeekend considerando l'ordine
    RigheRaggruppate= Table.Group(Origine,
                                  {"WeekdayOrWeekend"},
                                  {{"Conteggio Righe", each Table.RowCount(_), type number}},
                                  GroupKind.Local)

in
    RigheRaggruppate
```

	WeekdayOrWeekend	Conteggio Righe
1	Weekday	3
2	Weekend	2
3	Weekday	5
4	Weekend	2

M: ORDINARE I DATI DELLE TABELLE

VenditeGiornaliere

Data	WeekdayOrWeekend	Importo	Quantità
01-gen-14	Weekday	5	2
02-gen-14	Weekday	4	1
03-gen-14	Weekday	2	1
04-gen-14	Weekend	6	1
05-gen-14	Weekend	8	5
06-gen-14	Weekday	6	3
07-gen-14	Weekday	4	2
08-gen-14	Weekday	5	1
09-gen-14	Weekday	5	2
10-gen-14	Weekday	2	3
11-gen-14	Weekend	1	1
12-gen-14	Weekend	3	1

```
let
    //Carica i dati dalla tabella VenditeGiornaliere
    Origine= Excel.CurrentWorkbook(){[Name="VenditeGiornaliere"]}[Content],

    /* Ordina le righe per il campo WeekdayOrWeekend decrscente e
                        per il campo Importo crescente */
    RigheOrdinate= Table.Sort(Origine,{{"WeekdayOrWeekend", Order.Descending},
                                       {"Importo", Order.Ascending}})
in
    RigheOrdinate
```

	Data	WeekdayOrWeekend	Importo	Quantità
1	11/01/2014 00:00:00	Weekend	1	1
2	12/01/2014 00:00:00	Weekend	3	1
3	04/01/2014 00:00:00	Weekend	6	1
4	05/01/2014 00:00:00	Weekend	8	5
5	03/01/2014 00:00:00	Weekday	2	1
6	10/01/2014 00:00:00	Weekday	2	3
7	07/01/2014 00:00:00	Weekday	4	2
8	02/01/2014 00:00:00	Weekday	4	1
9	01/01/2014 00:00:00	Weekday	5	2
10	09/01/2014 00:00:00	Weekday	5	2
11	08/01/2014 00:00:00	Weekday	5	1
12	06/01/2014 00:00:00	Weekday	6	3

M: FILTRARE I DATI DELLE TABELLE

VenditeGiornaliere

Data	WeekdayOrWeekend	Importo	Quantità
01-gen-14	Weekday	5	2
02-gen-14	Weekday	4	1
03-gen-14	Weekday	2	1
04-gen-14	Weekend	6	1
05-gen-14	Weekend	8	5
06-gen-14	Weekday	6	3
07-gen-14	Weekday	4	2
08-gen-14	Weekday	5	1
09-gen-14	Weekday	5	2
10-gen-14	Weekday	2	3
11-gen-14	Weekend	1	1
12-gen-14	Weekend	3	1

```
let

    //Carica i dati dalla tabella VenditeGiornaliere
    Origine= Excel.CurrentWorkbook(){[Name="VenditeGiornaliere"]}[Content],

    //Filtra le righe con importo > 5
    RigheFiltrate= Table.SelectRows(Origine, each [Importo] > 5)
in
    RigheFiltrate
```

```
= Table.SelectRows(Origine, each [Importo] > 5)
```

	Data	WeekdayOrWeekend	Importo	Quantità
1	04/01/2014 00:00:00	Weekend	6	1
2	05/01/2014 00:00:00	Weekend	8	5
3	06/01/2014 00:00:00	Weekday	6	3

M: PIVOT/UNPIVOT I DATI DELLE TABELLE

VenditeGiornaliere

Data	WeekdayOrWeekend	Importo	Quantità
01-gen-14	Weekday	5	2
02-gen-14	Weekday	4	1
03-gen-14	Weekday	2	1
04-gen-14	Weekend	6	1
05-gen-14	Weekend	8	5
06-gen-14	Weekday	6	3
07-gen-14	Weekday	4	2
08-gen-14	Weekday	5	1
09-gen-14	Weekday	5	2
10-gen-14	Weekday	2	3
11-gen-14	Weekend	1	1
12-gen-14	Weekend	3	1

```
let

    //Carica i dati dalla tabella VenditeGiornaliere
    Origine= Excel.CurrentWorkbook(){[Name="VenditeGiornaliere"]}[Content],

    /*Unpivot le colonne Importo e Quantità
    ossia crea una colonna attributo che riporta l'intestazione (Importo e Quantità)
    e una colonna valore con i valori corrispondenti */

    Unpivot = Table.UnpivotOtherColumns(Origine,{"Data", "WeekdayOrWeekend"},"Attributo","Valori")

in
    Unpivot
```

	Data	WeekdayOrWeekend	Attributo	Valori
1	01/01/2014 00:00:00	Weekday	Importo	5
2	01/01/2014 00:00:00	Weekday	Quantità	2
3	02/01/2014 00:00:00	Weekday	Importo	4
4	02/01/2014 00:00:00	Weekday	Quantità	1
5	03/01/2014 00:00:00	Weekday	Importo	2
6	03/01/2014 00:00:00	Weekday	Quantità	1
7	04/01/2014 00:00:00	Weekend	Importo	6
8	04/01/2014 00:00:00	Weekend	Quantità	1
9	05/01/2014 00:00:00	Weekend	Importo	8
10	05/01/2014 00:00:00	Weekend	Quantità	5
11	06/01/2014 00:00:00	Weekday	Importo	6
12	06/01/2014 00:00:00	Weekday	Quantità	3
13	07/01/2014 00:00:00	Weekday	Importo	4
14	07/01/2014 00:00:00	Weekday	Quantità	2
15	08/01/2014 00:00:00	Weekday	Importo	5
16	08/01/2014 00:00:00	Weekday	Quantità	1
17	09/01/2014 00:00:00	Weekday	Importo	5
18	09/01/2014 00:00:00	Weekday	Quantità	2
19	10/01/2014 00:00:00	Weekday	Importo	2
20	10/01/2014 00:00:00	Weekday	Quantità	3
21	11/01/2014 00:00:00	Weekend	Importo	1
22	11/01/2014 00:00:00	Weekend	Quantità	1
23	12/01/2014 00:00:00	Weekend	Importo	3
24	12/01/2014 00:00:00	Weekend	Quantità	1

M: PIVOT/UNPIVOT I DATI DELLE TABELLE

VenditeGiornaliere

Data	WeekdayOrWeekend	Importo	Quantità
01-gen-14	Weekday	5	2
02-gen-14	Weekday	4	1
03-gen-14	Weekday	2	1
04-gen-14	Weekend	6	1
05-gen-14	Weekend	8	5
06-gen-14	Weekday	6	3
07-gen-14	Weekday	4	2
08-gen-14	Weekday	5	1
09-gen-14	Weekday	5	2
10-gen-14	Weekday	2	3
11-gen-14	Weekend	1	1
12-gen-14	Weekend	3	1

```
let

    //Carica i dati dalla tabella VenditeGiornaliere
    Origine= Excel.CurrentWorkbook(){[Name="VenditeGiornaliere"]}[Content],

    /* Pivotta il campo WeekdayOrWeekend
    ossia colonne separate per ogni occorrenza e somma gli importi
    se non ci sono importi inserisce null */

    PivottedTable = Table.Pivot(Origine, List.Distinct(Table.Column(Origine, "WeekdayOrWeekend")),
                                "WeekdayOrWeekend", "Importo", List.Sum)

in
    PivottedTable
```

	Data	Quantità	Weekday	Weekend
1	01/01/2014 00:00:00	2	5	null
2	02/01/2014 00:00:00	1	4	null
3	03/01/2014 00:00:00	1	2	null
4	04/01/2014 00:00:00	1	null	6
5	05/01/2014 00:00:00	5	null	8
6	06/01/2014 00:00:00	3	6	null
7	07/01/2014 00:00:00	2	4	null
8	08/01/2014 00:00:00	1	5	null
9	09/01/2014 00:00:00	2	5	null
10	10/01/2014 00:00:00	3	2	null
11	11/01/2014 00:00:00	1	null	1
12	12/01/2014 00:00:00	1	null	3

M: RIFERIMENTI IN UNA LISTA

- Fare un **riferimento** significa richiamare una valore
- Per fare riferimento ad un elemento di una lista occorre ricordarsi che la lista ha un ordine intrinseco e ci si può riferire ad un elemento in base ad un indice a base zero.

Quindi: per una lista {«A», «B», «C»}

L'espressione in M	{«A», «B», «C»}{0}	restituisce	«A»
L'espressione in M	{«A», «B», «C»}{2}	restituisce	«C»
L'espressione in M	{«A», «B», «C»}{4}	restituisce	Errore
L'espressione in M	{«A», «B», «C»}{4}?	restituisce	null

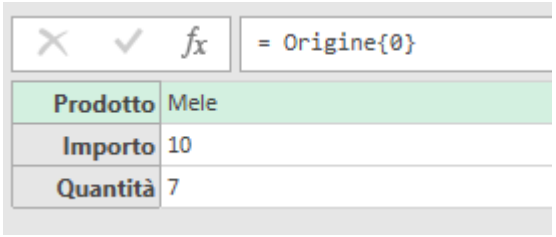
M: RIFERIMENTO AD UNA TABELLA

TabellaVendite

Prodotto	Importo	Quantità
Mele	10	7
Banane	20	8
Kiwi	30	9

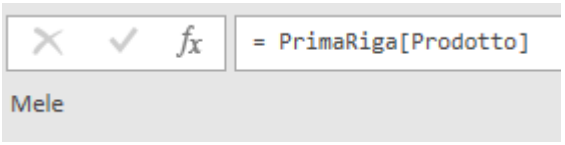
Riferimento alla prima riga

```
let
    //carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //estrae la prima riga
    PrimaRiga = Origine{0}

in
    PrimaRiga
```



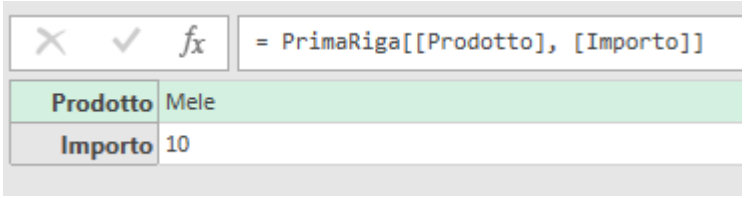
Riferimento all'incrocio tra prima riga e prima colonna

```
let
    //carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //estrae la prima riga
    PrimaRiga = Origine{0}?,

    //restituisce il vaole dalla prima colonna della prima riga, denominata Prodotto
    PrimaColonna = PrimaRiga[Prodotto]

in
    PrimaColonna
```



Riferimento all'incrocio tra prima riga e prima colonna, riportando anche l'importo

```
let
    //carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //estrae la prima riga
    PrimaRiga = Origine{0}?,

    //restituisce il valore dalla prima colonna della prima riga, denominata Prodotto e ralativo importo
    PrimaColonna = PrimaRiga[[Prodotto], [Importo]]

in
    PrimaColonna
```



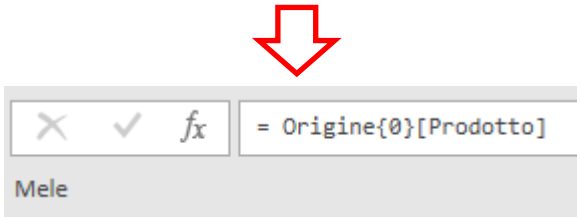
M: RIFERIMENTO AD UNA TABELLA

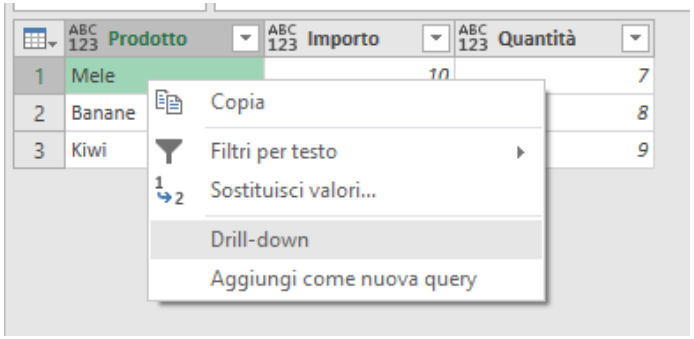
TabellaVendite

Prodotto	Importo	Quantità
Mele	10	7
Banane	20	8
Kiwi	30	9

Riferimento ad uno specifico valore della tabella

```
let
    //carica i dati dalla tabella TabellaVendite
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //restituisce il valore dalla prima colonna della prima riga, denominata Prodotto
    PrimoValore = Origine{0}[Prodotto]

in
    PrimoValore
```





E' possibile ottenere lo stesso risultato senza scrivere codice ma utilizzando le funzionalità dell'interfaccia utente.

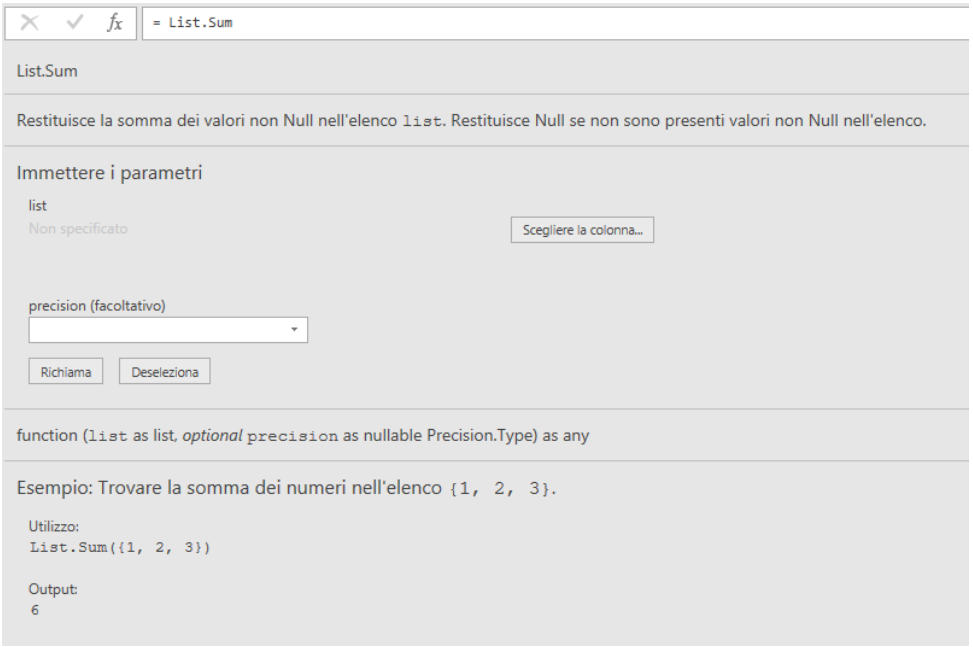
Dx mouse sul valore che si intende estrarre > Drill-down

M: LE FUNZIONI

- Sono numerosissime le funzioni che possono essere utilizzate in Power Query
- La libreria standard è consultabile al seguente link, ed è in costante aggiornamento:

  https://msdn.microsoft.com/query-bi/m/power-query-m-function-reference

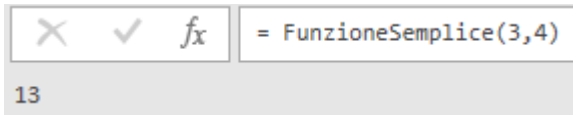
- Per ogni funzione (raggruppate in categorie) è definita la sintassi e sono forniti degli esempi
- Digitando una funzione nella barra della formula (senza la parentesi tonda) si aprirà un tutorial (creazione guidata per il corretto inserimento della funzione). Ad esempio per la funzione List.Sum:



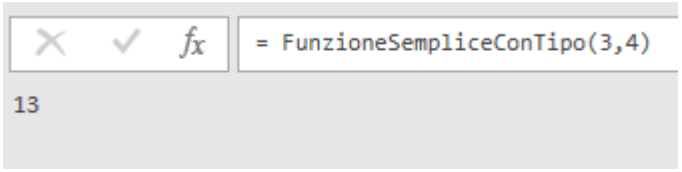
- E' possibile tuttavia definire le proprie funzioni

M: LE FUNZIONI PERSONALIZZATE

```
let
    //Definisce una funzione personalizzata FunzioneSemplice che moltiplica due valori e aggiunge 1
    FunzioneSemplice = (x,y) => (x * y) +1,

    //Richiama la funzione con i parametri 3 e 4
    Risultato= FunzioneSemplice(3,4)

in
    Risultato
```



```
let
    /*Definisce una funzione personalizzata FunzioneSemplice che moltiplica due valori
      e aggiunge un terzo parametro (opzionale se non specificato è uguale a 1)
      per i parametri specifica i Tipo Valore */

    FunzioneSempliceConTipo= (x as number,y as number, optional z as number)
                             as number => (x * y) + (if z=null then 1 else z),

    //Richiama la funzione con i parametri 3 e 4
    Risultato= FunzioneSempliceConTipo(3,4)
in
    Risultato
```



M: LE FUNZIONI PERSONALIZZATE

```
let
    //Definisce una funzione che moltiplica un numero qualsiasi per 2

    EsempioFunzioneEach = each _ * 2,

    //Richiama la funzione con il parametro 3
    Risultato = EsempioFunzioneEach(3)

in
    Risultato
```

L'espressione *each* può essere utilizzata ogni volta che viene dichiarata una funzione. E' una scorciatoia per definire un singolo parametro senza Tipo Valore denominato _

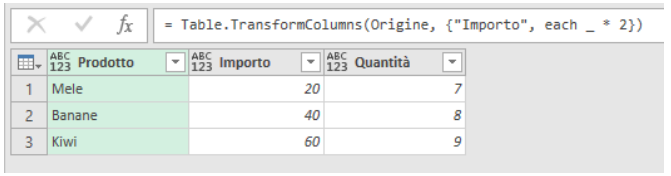
In questo caso l'espressione *each* _*2 è equivalente a (_) => _*2

Più spesso l'espressione *each* viene utilizzata per passare parametri ad altre funzioni.

TabellaVendite

Prodotto	Importo	Quantità
Mele	10	7
Banane	20	8
Kiwi	30	9

```
let
    //Carica i dati dalla tabella TabellaVendite
    Origine= Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Moltiplica ogni valore di Importo per 2
    ImportoPerDue= Table.TransformColumns(Origine, {"Importo", each _ * 2})

in
    ImportoPerDue
```

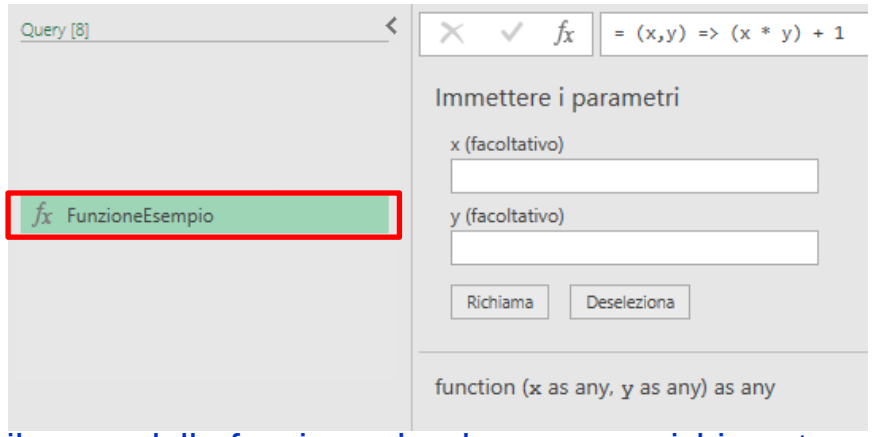


	Prodotto	Importo	Quantità
1	Mele	20	7
2	Banane	40	8
3	Kiwi	60	9

M: CREARE UNA QUERY CHE FUNGA DA FUNZIONE

FunzioneEsempio

```
let
    //Definisco la funzione
    FunzionePersonalizzata= (x,y) => (x * y) + 1

in
    FunzionePersonalizzata
```



Il nome della Query (non quella nel codice) definisce il nome della funzione che deve essere richiamato (in questo caso FunzioneEsempio). Una funzione non da risultati fino a quando non viene richiamata.

TabellaVendite

Prodotto	Importo	Quantità
Mele	10	7
Banane	20	8
Kiwi	30	9

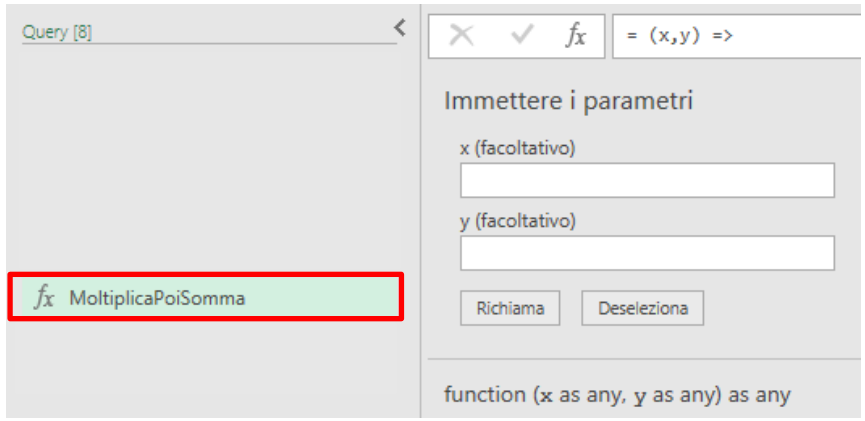
```
let
    //Carica i dati dalla tabella TabellaVendite
    Origine= Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Richiama la funzione personalizzata in una nuova colonna
    InserisciColonnaPersonalizzata = Table.AddColumn(Origine,
                                     "FunzioneEsempio",
                                     each FunzioneEsempio([Importo],[Quantità]))
in
    InserisciColonnaPersonalizzata
```

	Prodotto	Importo	Quantità	FunzioneEsempio
1	Mele	10	7	71
2	Banane	20	8	161
3	Kiwi	30	9	271

M: FUNZIONI PERSONALIZZATE PIU' COMPLESSE

MoltiplicaPoiSomma

```
let
    //Definisce una funzione con due passaggi attraverso un'espressione Let

    FunzionePersonalizzata = (x,y) =>
      let
          //Moltiplica x e y
          Passo1= x * y,

          //Aggiunge 1 al risultato del Passo1
          Passo2 = Passo1 + 1
      in
          Passo2

in
    FunzionePersonalizzata
```



In questo caso la funzione viene definita in due passaggio attraverso un'espressione *let ... in*

Una funzione non da risultati fino a quando non viene richiamata.

TabellaVendite

Prodotto	Importo	Quantità
Mele	10	7
Banane	20	8
Kiwi	30	9

```
let
    //Carica i dati dalla tabella TabellaVendite
    Origine= Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Richiama la funzione personalizzata in una nuova colonna
    InserisciColonnaPersonalizzata = Table.AddColumn(Origine,
                                     "MoltiplicaPoiSomma",
                                     each MoltiplicaPoiSomma([Importo],[Quantità]))
in
    InserisciColonnaPersonalizzata
```

	Prodotto	Importo	Quantità	MoltiplicaPoiSomma
1	Mele	10	7	71
2	Banane	20	8	161
3	Kiwi	30	9	271

M: FUNZIONI RICORSIVE

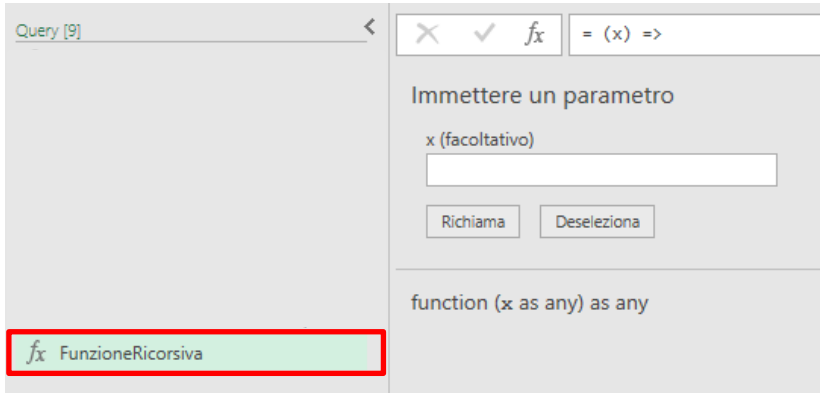
Una funzione ricorsiva è una funzione che richiama se stessa se una certa condizione si verifica (onde evitare che il richiamo vada avanti all'infinito).

Il processo ricorsivo avviene mediante l'espressione *if...then...else*

La funzione nel codice viene richiamata con l'operatore @

FunzioneRicorsiva

```
let
    //Definisce una funzione ricorsiva con un parametro

    RaddoppiaFinoACento = (x) =>

      //se il parametro x è > o = a 100 restituisce il parametro
      if x > 100
       then x

      //in caso contrario richiama la funzione originale con x * 2
       else @RaddoppiaFinoACento(x*2)

in
    RaddoppiaFinoACento
```



Se si inserisce il parametro 4, la funzione restituirà 128

M: TRASFERIRE UNA QUERY TRA DUE FILE

Per trasferire una query da un file ad un altro, nelle prime versioni di Power Query, era necessario aprire l'Editor Avanzato, copiare il codice ed incollarlo in una query vuota della nuova cartella di lavoro.

Nelle versioni più recenti (Microsoft 365) è sufficiente copiare (CTRL+C) la query dal riquadro Query e Connessioni e incollare (CTRL+V) nel riquadro del file di destinazione (ovviamente occorrerà aggiornare i percorsi di origine)

Guarda il video tutorial di questa funzionalità su YouTube:
EXCEL - POWER QUERY: Copiare e trasferire una query da un file ad un altro (link)

POWER QUERY: LAVORARE CON PIU' QUERY

USARE UNA QUERY COME FONTE DI UN'ALTRA

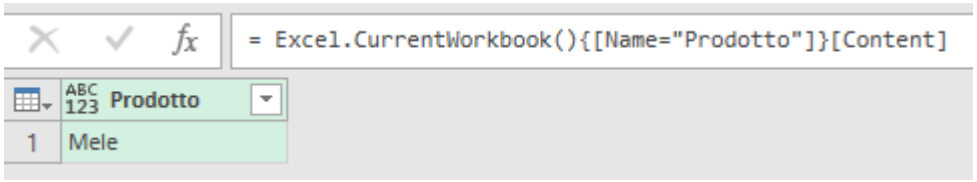
- Per fare riferimento all'output di una query esistente come origine di una nuova query, è sufficiente cliccare **dx mouse sul nome della query nel Pannello di Navigazione e selezionare il comando riferimento**. Verrà creata una nuova query che potrà essere rinominata:

ProdottoQuery

Prodotto

Prodotto
Mele

```
let
    Origine = Excel.CurrentWorkbook(){[Name="Prodotto"]}[Content]
in
    Origine
```



ProdottoQuery (2)

Nel codice M è sufficiente riferirsi al nome della query di origine

```
let
    Origine = ProdottoQuery
in
    Origine
```

CREARE UNA QUERY PARAMETRIZZATA

- Un query parametrizzata con due fonti dati (basedati e parametro). Il parametro fornisce le informazioni relativamente, ad esempio, al filtro da applicare alla basedati.

Basedati

TabellaVendite

Mese	Prodotto	Importo
Gennaio	Mele	1
Febbraio	Mele	2
Marzo	Mele	3
Gennaio	Pere	10
Febbraio	Pere	12
Marzo	Pere	14
Gennaio	Kiwi	20
Febbraio	Kiwi	23
Marzo	Kiwi	26

Parametro

Prodotto

Prodotto
Mele

La query ProductQuery (slide precedente) carica la tabella Prodotto (in cui l'utente inserirà il parametro – nell'esempio «Mele»)

VenditeParametro Codice se filtro in base a parametro

```
let
    //Carica la tabella TabellaVendite dalla cartella di lavoro
    Origine= Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Filtra le righe rispetto al parametro inserito nella prima riga della ProdottoQuery
    RigheFiltrate= Table.SelectRows(Origine, each ([Prodotto] = ProdottoQuery[Prodotto]{0}))

in
    RigheFiltrate
```

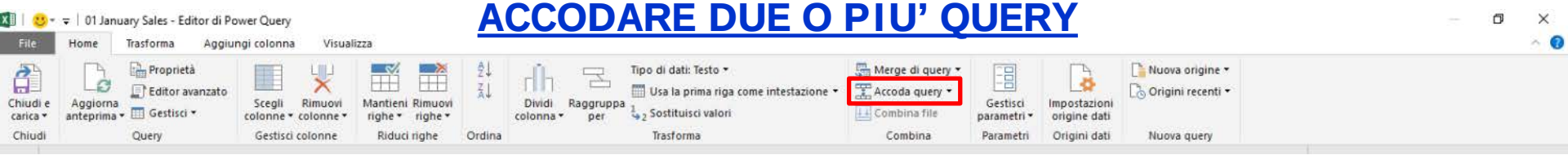
VenditeMele Codice se filtro «Mele»

```
let
    //Carica la tabella TabellaVendite dalla cartella di lavoro
    Origine= Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],

    //Filtra le righe per il prodotto "Mele"
    RigheFiltrate= Table.SelectRows(Origine, each ([Prodotto] = "Mele"))

in
    RigheFiltrate
```

Il risultato è lo stesso (a meno che cambi il parametro)

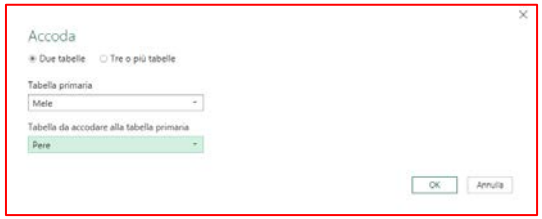
	Mese	Prodotto	Importo
1	Gennaio	Mele	1
2	Febbraio	Mele	2
3	Marzo	Mele	3

ACCODARE DUE O PIU' QUERY



- Per accodare (ossia trasformare in un unico elenco due o più query) occorre selezionare **Home > Combina > Accoda Query > Accoda query** oppure **Accoda query come nuove** (a seconda che si voglia creare una nuova query o accodare a quella esistente)

Due tabelle



	Mese	Prodotto	Quantità
1	Gennaio	Mele	1
2	Febbraio	Mele	2
3	Marzo	Mele	3
4	Gennaio	Pere	5
5	Febbraio	Pere	7
6	Marzo	Pere	9

```
let
    Origine = Table.Combine({Mele, Pere})
in
    Origine
```

Mele

Mese	Prodotto	Quantità
Gennaio	Mele	1
Febbraio	Mele	2
Marzo	Mele	3

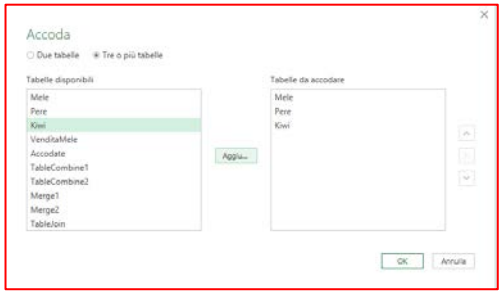
Pere

Mese	Prodotto	Quantità
Gennaio	Pere	5
Febbraio	Pere	7
Marzo	Pere	9

Kiwi

Mese	Prodotto	Quantità
Gennaio	Kiwi	20
Febbraio	Kiwi	23
Marzo	Kiwi	26

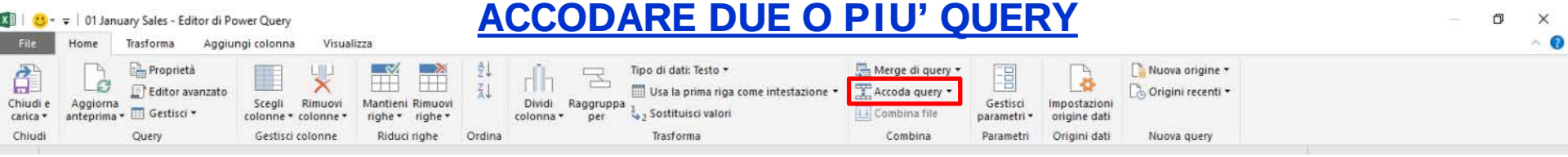
Tre o più tabelle



	Mese	Prodotto	Quantità
1	Gennaio	Mele	1
2	Febbraio	Mele	2
3	Marzo	Mele	3
4	Gennaio	Pere	5
5	Febbraio	Pere	7
6	Marzo	Pere	9
7	Gennaio	Kiwi	20
8	Febbraio	Kiwi	23
9	Marzo	Kiwi	26

```
let
    Origine = Table.Combine({Mele, Pere, Kiwi})
in
    Origine
```

ACCODARE DUE O PIU' QUERY



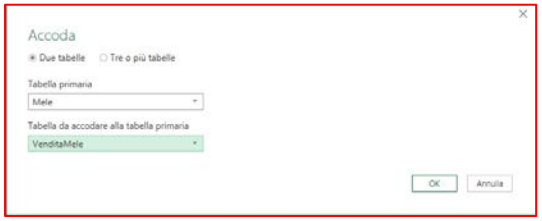
- Per accodare (ossia trasformare in un unico elenco due o più query) occorre selezionare **Home > Combina > Accoda Query > Accoda query** oppure **Accoda query come nuove** (a seconda che si voglia creare una nuova query o accodare a quella esistente)
- Se si prova ad accodare tabelle con campi diversi:

Mele

Mese	Prodotto	Quantità
Gennaio	Mele	1
Febbraio	Mele	2
Marzo	Mele	3

VenditaMele

Mese	Prodotto	Importo
Gennaio	Mele	100
Febbraio	Mele	200
Marzo	Mele	300




```
let
    Origine = Table.Combine({Mele, VenditaMele})
in
    Origine
```

	Mese	Prodotto	Quantità	Importo
1	Gennaio	Mele	1	null
2	Febbraio	Mele	2	null
3	Marzo	Mele	3	null
4	Gennaio	Mele	null	100
5	Febbraio	Mele	null	200
6	Marzo	Mele	null	300

MERGE DI DUE QUERY

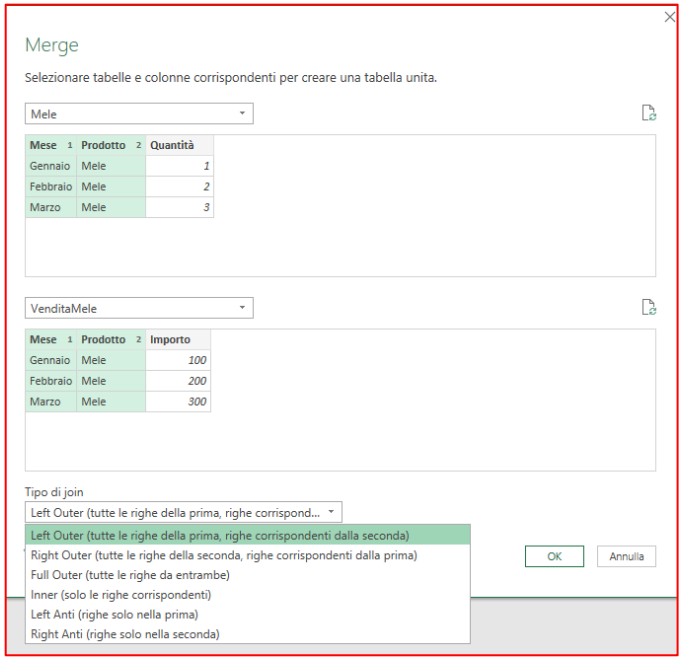
- Per accodare (ossia trasformare in un unico elenco due o più query) occorre selezionare **Home > Combina > Merge di query > Merge di query** oppure **Unisci query come nuova** (a seconda che si voglia creare una nuova query o accodare a quella esistente)

Mele

Mese	Prodotto	Quantità
Gennaio	Mele	1
Febbraio	Mele	2
Marzo	Mele	3

VenditaMele

Mese	Prodotto	Importo
Gennaio	Mele	100
Febbraio	Mele	200
Marzo	Mele	300




```
let
    Origine = Table.NestedJoin(Mele,
                {"Mese", "Prodotto"},VenditaMele,
                {"Mese", "Prodotto"},"VenditaMele",
                JoinKind.LeftOuter)
in
    Origine
```

NB: Occorre selezionare le colonne che devono essere abbinate nelle due tabelle (nello stesso ordine). Power Query non fa nessun controllo sulle intestazioni

	Mese	Prodotto	Quantità	VenditaMele
1	Gennaio	Mele	1	Table
2	Febbraio	Mele	2	Table
3	Marzo	Mele	3	Table

MERGE DI DUE QUERY

- Per accodare (ossia trasformare in un unico elenco due o più query) occorre selezionare **Home > Combina > Merge di query > Merge di query** oppure **Unisci query come nuova** (a seconda che si voglia creare una nuova query o accodare a quella esistente)
- La tabella restituita presenta una nuova colonna con i campi della tabella VenditaMele che può essere espansa cliccando sulle due frecce nell'intestazione (si potranno scegliere i campi).

Mele

Mese	Prodotto	Quantità
Gennaio	Mele	1
Febbraio	Mele	2
Marzo	Mele	3

VenditaMele

Mese	Prodotto	Importo
Gennaio	Mele	100
Febbraio	Mele	200
Marzo	Mele	300

	Mese	Prodotto	Quantità	VenditaMele
1	Gennaio	Mele	1	Table
2	Febbraio	Mele	2	Table
3	Marzo	Mele	3	Table

Se si sceglie solo Importo

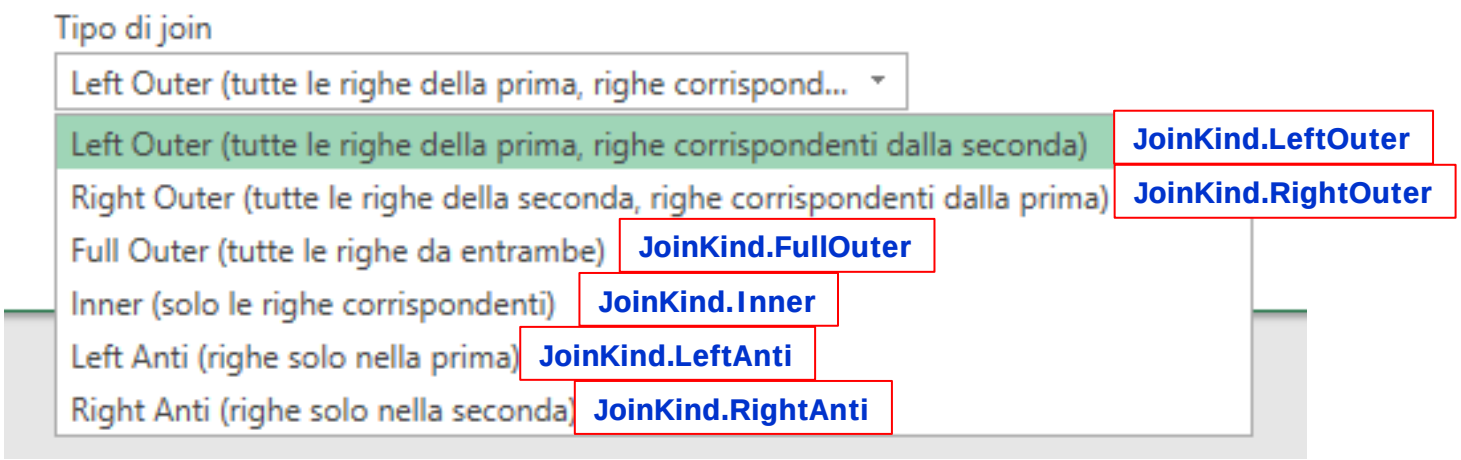
	Mese	Prodotto	Quantità	VenditaMele.Importo
1	Gennaio	Mele	1	100
2	Febbraio	Mele	2	200
3	Marzo	Mele	3	300

```
let
    Origine = Table.NestedJoin(Mele,
              {"Mese", "Prodotto"},VenditaMele,
              {"Mese", "Prodotto"},"VenditaMele",
              JoinKind.LeftOuter),
    #"Tabella VenditaMele espansa" = Table.ExpandTableColumn(
                                     Origine, "VenditaMele",
                                     {"Importo"},
                                     {"VenditaMele.Importo"})
in
    #"Tabella VenditaMele espansa"
```

NB: In fase di espansione è possibile scegliere l'opzione aggrega (che permette di fare dei calcoli)

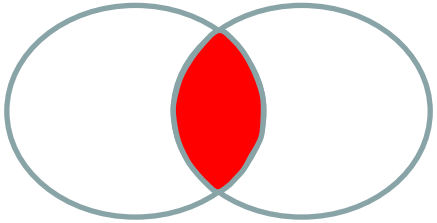
MERGE DI DUE QUERY

- Per accodare (ossia trasformare in un unico elenco due o più query) occorre selezionare **Home > Combina > Merge di query > Merge di query** oppure **Unisci query come nuova** (a seconda che si voglia creare una nuova query o accodare a quella esistente)
- Esistono diversi tipi di join (che corrispondono ad altrettante parametri della funzione Table.NestedJoin() nel codice M) a seconda dell'output desiderato.

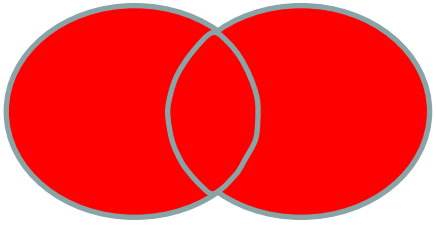


MERGE DI DUE QUERY

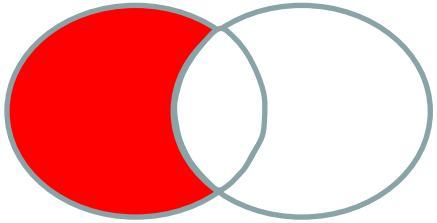
JoinKind.Inner



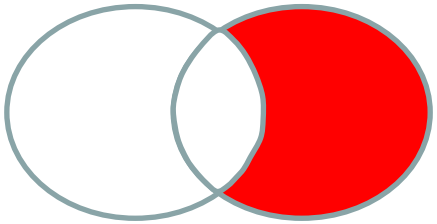
JoinKind.FullOuter



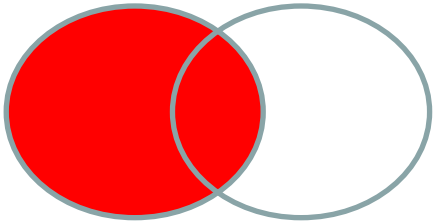
JoinKind.LeftAnti



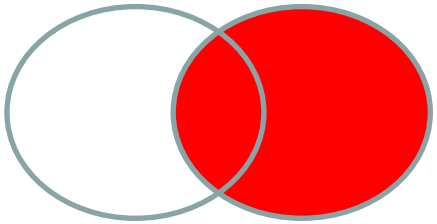
JoinKind.RightAnti



JoinKind.LeftOuter



JoinKind.RightOuter



POWER QUERY: ALCUNI ESERCIZI

Esercizio 1:
MS Excel - POWER QUERY:
Calcolare la percentuali di una colonna

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Prodotto", type text}, {"Importo", Int64.Type}}),
    #"Raggruppate righe" = Table.Group(#"Modificato tipo", {}, {{"Totale", each List.Sum([Importo]), type number}}),
    Totale = #"Raggruppate righe"{0}[Totale],
    Personalizzato1 = Origine,
    #"Aggiunta colonna personalizzata" = Table.AddColumn(Personalizzato1, "Quota", each [Importo]/Totale),
    #"Modificato tipo1" = Table.TransformColumnTypes(#"Aggiunta colonna personalizzata",{{"Quota", Percentage.Type}})
in
    #"Modificato tipo1"
```

Esercizio 2:
MS Excel - POWER QUERY:
Calcolare la percentuali sulla riga precedente

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Data", type datetime}, {"Importo", type number}}),
    #"Aggiunta colonna personalizzata" = Table.AddColumn(#"Modificato tipo", "Giorno Precedente", each [Data]-#duration(1,0,0,0)),
    #"Merge di query eseguito" = Table.NestedJoin(#"Aggiunta colonna personalizzata",{"Giorno Precedente"},
        #"Aggiunta colonna personalizzata",{"Data"},"Aggiunta colonna personalizzata",JoinKind.LeftOuter),
    #"Tabella Aggiunta colonna personalizzata espansa" = Table.ExpandTableColumn(#"Merge di query eseguito",
        "Aggiunta colonna personalizzata", {"Importo"}, {"Aggiunta colonna personalizzata.Importo"}),
    #"Rinominate colonne" = Table.RenameColumns(#"Tabella Aggiunta colonna personalizzata espansa",
        {{"Aggiunta colonna personalizzata.Importo", "Importo Giorno Prec"}}),
    #"Aggiunta colonna personalizzata1" = Table.AddColumn(#"Rinominate colonne",
        "Crescita Percentuale", each ([Importo]-[Importo Giorno Prec])/[Importo Giorno Prec]),
    #"Modificato tipo1" = Table.TransformColumnTypes(#"Aggiunta colonna personalizzata1",{{"Crescita Percentuale", Percentage.Type}}),
    #"Rimosse colonne" = Table.RemoveColumns(#"Modificato tipo1",{"Giorno Precedente", "Importo Giorno Prec"}),
    #"Ordinate righe" = Table.Sort(#"Rimosse colonne",{{"Data", Order.Ascending}})
in
    #"Ordinate righe"
```

Esercizio 3:
MS Excel - POWER QUERY:
Il database sempre ordinato (inserire il Rango)

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="TabellaVendite"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Prodotto", type text}, {"Importo", Int64.Type}}),
    Rango = (Vendite) => Table.RowCount(Table.SelectRows(Origine, each [Importo]>Vendite))+1,
    Personalizzato2 = Origine,
    #"Aggiunta colonna personalizzata" = Table.AddColumn(Personalizzato2, "Rango", each Rango([Importo])),
    #"Ordinate righe" = Table.Sort(#"Aggiunta colonna personalizzata",{{"Rango", Order.Ascending}})
in
    #"Ordinate righe"
```

Esercizio 4:
MS Excel - POWER QUERY:
Conteggio righe univoche (distinct count)

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="ContegioClientiUnivoci"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Data", type date}, {"Cliente", type text}, {"Sales", type number}}),
    #"Rimossi duplicati" = Table.Distinct(#"Modificato tipo", {"Data", "Cliente"}),
    #"Raggruppate righe" = Table.Group(#"Rimossi duplicati", {"Data"}, {{"Clienti Univoci", each Table.RowCount(_), type number}})
in
    #"Raggruppate righe"
```

Esercizio 5:
MS Excel - POWER QUERY:
Trasformare un elenco in una tabella

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="Lista"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Informazioni", type text}}),
    #"Aggiunta colonna indice" = Table.AddIndexColumn(#"Modificato tipo", "Indice", 0, 1),
    #"Inserita operazione modulo" = Table.AddColumn(#"Aggiunta colonna indice",
        "Modulo", each Number.Mod([Indice], 3), type number),
    #"Modificato tipo1" = Table.TransformColumnTypes(#"Inserita operazione modulo",{{"Modulo", type text}}),
    #"Sostituito valore" = Table.ReplaceValue(#"Modificato tipo1","0","Nome",Replacer.ReplaceText,{"Modulo"}),
    #"Sostituito valore1" = Table.ReplaceValue(#"Sostituito valore","1","Genere",Replacer.ReplaceText,{"Modulo"}),
    #"Sostituito valore2" = Table.ReplaceValue(#"Sostituito valore1","2","Nazione",Replacer.ReplaceText,{"Modulo"}),
    #"Rinominate colonne" = Table.RenameColumns(#"Sostituito valore2",{{"Modulo", "Tipo Riga"}}),
    #"Inserita divisione intera" = Table.AddColumn(#"Rinominate colonne",
        "Divisione intera", each Number.IntegerDivide([Indice], 3), Int64.Type),
    #"Rinominate colonne1" = Table.RenameColumns(#"Inserita divisione intera",{{"Divisione intera", "Codice Cliente"}}),
    #"Rimosse colonne" = Table.RemoveColumns(#"Rinominate colonne1",{"Indice"}),
    Personalizzato1 = Table.Pivot(#"Rimosse colonne",{"Nome", "Genere", "Nazione"},"Tipo Riga","Informazioni")
in
    Personalizzato1
```

Esercizio 6:
MS Excel - POWER QUERY:
Confrontare due elenchi

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Table.Combine({ANNOClienti, APClienti}),
    #"Merge di colonne" = Table.CombineColumns(Origine,{"Clienti Attuali", "Clienti Scorso Anno"},
            Combiner.CombineTextByDelimiter("", QuoteStyle.None),"Sottoposto a merge"),
    #"Rinominate colonne" = Table.RenameColumns(#"Merge di colonne",{{"Sottoposto a merge", "Clienti"}}),
    #"Rimossi duplicati" = Table.Distinct(#"Rinominate colonne"),
    #"Merge di query eseguito" = Table.NestedJoin(#"Rimossi duplicati",{"Clienti"},ANNOClienti,
        {"Clienti Attuali"},"ANNOClienti",JoinKind.LeftOuter),
    #"Colonne ANNOClienti aggregate" = Table.AggregateTableColumn(#"Merge di query eseguito", "ANNOClienti",
        {{"Clienti Attuali", List.NonNullCount, "Conteggio (non vuoto) di ANNOClienti.Clienti Attuali"}}),
    #"Rinominate colonne1" = Table.RenameColumns(#"Colonne ANNOClienti aggregate",
        {{"Conteggio (non vuoto) di ANNOClienti.Clienti Attuali", "Attuali"}}),
    #"Merge di query eseguito1" = Table.NestedJoin(#"Rinominate colonne1",{"Clienti"},APClienti,
        {"Clienti Scorso Anno"},"APClienti",JoinKind.LeftOuter),
    #"Colonne APClienti aggregate" = Table.AggregateTableColumn(#"Merge di query eseguito1", "APClienti",
        {{"Clienti Scorso Anno", List.NonNullCount, "Conteggio (non vuoto) di APClienti.Clienti Scorso Anno"}}),
    #"Rinominate colonne2" = Table.RenameColumns(#"Colonne APClienti aggregate",
        {{"Conteggio (non vuoto) di APClienti.Clienti Scorso Anno", "Passati"}}),
    #"Aggiunta colonna personalizzata" = Table.AddColumn(#"Rinominate colonne2", "Classificazione",
        each if [Attuali]=1 and [Passati] =1 then "Fidelizzato" else
        if [Attuali]=1 and [Passati] =0 then "Nuovo"
        else "Perso"),
    #"Rimosse colonne" = Table.RemoveColumns(#"Aggiunta colonna personalizzata",{"Attuali", "Passati"}),
    #"Ordinate righe" = Table.Sort(#"Rimosse colonne",{{"Clienti", Order.Ascending}})
in
    #"Ordinate righe"
```

Esercizio 7:
MS Excel - POWER QUERY:
Generare una tabella temporale (calendario)

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = List.Dates(#date(2018,1,1),365,#duration(1,0,0,0)),
    #"Conversione in tabella" = Table.FromList(Origine, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    #"Rinominate colonne" = Table.RenameColumns(#"Conversione in tabella",{{"Column1", "Data"}}),
    #"Modificato tipo" = Table.TransformColumnTypes(#"Rinominate colonne",{{"Data", type date}}),
    #"Duplicata colonna" = Table.DuplicateColumn(#"Modificato tipo", "Data", "Data - Copia"),
    #"Rinominate colonne1" = Table.RenameColumns(#"Duplicata colonna",{{"Data - Copia", "Anno"}}),
    #"Anno estratto" = Table.TransformColumns(#"Rinominate colonne1",{{"Anno", Date.Year, Int64.Type}}),
    #"Aggiunta colonna personalizzata" = Table.AddColumn(#"Anno estratto", "Mese", each Date.ToText([Data], "MMMM")),
    #"Aggiunta colonna personalizzata1" = Table.AddColumn(#"Aggiunta colonna personalizzata",
        "Giorno Nome", each Date.ToText([Data], "dddd")),
    #"Inserita settimana dell'anno" = Table.AddColumn(#"Aggiunta colonna personalizzata1",
        "Settimana dell'anno", each Date.WeekOfYear([Data]), Int64.Type),
    #"Rinominate colonne2" = Table.RenameColumns(#"Inserita settimana dell'anno",{{"Settimana dell'anno", "Settimana"}})
in
    #"Rinominate colonne2"
```

Esercizio 8:
MS Excel - POWER QUERY:
Misurare la durata massima

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="PrezzoTitolo"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Tempo", type time}, {"Quotazione", type number}}),
    #"Aggiunta colonna personalizzata" = Table.AddColumn(#"Modificato tipo", "Sopra50", each [Quotazione]>=50),
    #"Modificato tipo1" = Table.TransformColumnTypes(#"Aggiunta colonna personalizzata",{{"Sopra50", type logical}}),
    #"Raggruppate righe" = Table.Group(#"Modificato tipo1", {"Sopra50"}, {{"Inizio", each List.Min([Tempo]), type time},
          {"Fine", each List.Max([Tempo]), type time}, {"Minuti", each Table.RowCount(_), type number}}, GroupKind.Local),
    #"Aggiunta colonna personalizzata1" = Table.AddColumn(#"Raggruppate righe", "Fine+1", each [Fine]+#duration(0,0,1,0)),
    #"Modificato tipo2" = Table.TransformColumnTypes(#"Aggiunta colonna personalizzata1",{{"Fine+1", type time}}),
    #"Rimosse colonne" = Table.RemoveColumns(#"Modificato tipo2",{"Fine"}),
    #"Rinominate colonne" = Table.RenameColumns(#"Rimosse colonne",{{"Fine+1", "Fine"}}),
    #"Riordinate colonne" = Table.ReorderColumns(#"Rinominate colonne",{"Sopra50", "Inizio", "Fine", "Minuti"}),
    #"Filtrate righe" = Table.SelectRows(#"Riordinate colonne", each ([Sopra50] = true)),
    #"Ordinate righe" = Table.Sort(#"Filtrate righe",{{"Minuti", Order.Descending}}),
    #"Mantenute prime righe" = Table.FirstN(#"Ordinate righe",1)
in
    #"Mantenute prime righe"
```

Esercizio 9:
MS Excel - POWER QUERY:
Elaborare info dal web - Previsioni Meteo

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice funzione:

```
let
    OttieniPrevisioni = (Città as text) as table =>

    let
        Origine = Web.Page(Web.Contents("https://www.ilmeteo.it/meteo/" & Città)),
           Data0 = Origine{0}[Data],
        #"Modificato tipo" = Table.TransformColumnTypes(Data0,{{"Ora", type time}, {"", type text},
            {"Tempo", type text}, {"T (°C)", type text}, {"2", type text}, {"Vento (km/h)Umidità", type text},
            {"PrecipitazioniGrandine", type text}, {"Quota 0°CPressione", type text},
            {"VisibilitàPercepita", type text}, {"3", type text}, {"U.R.UV", type text}}),
        #"Rimossi errori" = Table.RemoveRowsWithErrors(#"Modificato tipo", {"Ora"}),
        #"Rimosse colonne" = Table.RemoveColumns(#"Rimossi errori",{""}),
        #"Filtrate righe" = Table.SelectRows(#"Rimosse colonne",
            each ([Tempo] <> "Segnala il tempo, diventa Meteo Reporter!")),
        #"Rinominate colonne" = Table.RenameColumns(#"Filtrate righe",{{"Tempo", "Meteo"}}),
        #"Rimosse colonne1" = Table.RemoveColumns(#"Rinominate colonne",{"2", "Vento (km/h)Umidità",
            "PrecipitazioniGrandine", "Quota 0°CPressione", "VisibilitàPercepita", "3", "U.R.UV"})
    in
        #"Rimosse colonne1"

in
    OttieniPrevisioni
```

Esercizio 9:
MS Excel - POWER QUERY:
Elaborare info dal web - Previsioni Meteo

Guardalo su YouTube – il file è scaricabile dal commento del video

Codice tabella:

```
let
    Origine = Excel.CurrentWorkbook(){[Name="Città"]}[Content],
    #"Modificato tipo" = Table.TransformColumnTypes(Origine,{{"Città", type text}}),
    #"Convertita in maiuscolo ogni parola" = Table.TransformColumns(#"Modificato tipo",
        {{"Città", Text.Proper, type text}}),
    #"Aggiunta colonna personalizzata" = Table.AddColumn(#"Convertita in maiuscolo ogni parola",
        "Previsioni", each OttieniPrevisioni([Città])),
    #"Tabella Previsioni espansa" = Table.ExpandTableColumn(#"Aggiunta colonna personalizzata",
        "Previsioni", {"Ora", "Meteo", "T (°C)"}, {"Previsioni.Ora", "Previsioni.Meteo", "Previsioni.T (°C)"})
in
    #"Tabella Previsioni espansa"
```

Esercizio 10:
EXCEL- POWER QUERY:
Creare una FUNZIONE PERSONALIZZATA per il calcolo del SALDO PROGRESSIVO
Guardalo su YouTube – il file è scaricabile dal commento del video

Esercizio 11:
EXCEL - POWER QUERY:
Funzioni LIST.ACCUMULATE e LIST.SUM per il calcolo del SALDO PROGRESSIVO
Guardalo su YouTube – il file è scaricabile dal commento del video

Esercizio 12:
EXCEL - POWER QUERY:
Gestire la competenza dei costi
Guardalo su YouTube – il file è scaricabile dal commento del video

Esercizio 13:
EXCEL- POWER QUERY:
Importare dati da più file XML - Creare un database con le Fatture Elettroniche
Guardalo su YouTube – il file è scaricabile dal commento del video

Esercizio 14:
EXCEL - POWER QUERY:
LIST.FIRSTN e LIST.BUFFER per il calcolo del saldo progressivo di una colonna
Guardalo su YouTube – il file è scaricabile dal commento del video

Esercizio 15:
EXCEL - POWER QUERY:
Funzione personalizzata per il calcolo del progressivo parziale di una colonna
Guardalo su YouTube – il file è scaricabile dal commento del video

Esercizio 16:
EXCEL: La sfida #4:
Alternative al CERCA.VERT()
Guardalo su YouTube – il file è scaricabile dal commento del video

CONTATTI

Dott. EMMANUELE VIETTI – e.vietti@experta-bs.it

EXPERTA BUSINESS Solutions Srl
Via Filangieri n. 16 10128 Torino
Tel.: +39 011 5183742
Fax: +39 011 19715613
Mail: info@experta-bs.it
Web: www.experta-bs.it

Partecipa al gruppo Linkedin:
UTILIZZO PROFESSIONALE DI MS-EXCEL (link)

Iscriviti al canale YouTube:
UTILIZZO PROFESSIONALE DI MICROSOFT OFFICE (link)

Iscriviti al canale Telegram:
UTILIZZO PROFESSIONALE DI MICROSOFT OFFICE (link)

Segui la pagina Facebook:
UTILIZZO PROFESSIONALE DI MICROSOFT EXCEL (link)