Obiettivo dell'Esercitazione

L'obiettivo di questa esercitazione è creare una pipeline utilizzando Azure Data Factory che consenta il caricamento di un file CSV su Azure Blob Storage, la pulizia e la rimappatura delle colonne in italiano. Il processo deve mantenere solo le colonne 'Film', 'Generi' e 'Valutazioni', filtrare i film con valutazioni superiori a 7 e garantire un trasferimento dati efficiente e parallelo. Inoltre, è essenziale conservare i metadati per eventuali informazioni aggiuntive utili.

Svolgimento dell'Esercitazione

1. Impostazione del Resource Group, Managed Identity e Key Vault

Dopo l'attivazione della sottoscrizione, è stato creato un Resource Group inserendo la sottoscrizione e la posizione desiderata (*Fig. 1*).
È stata creata una Managed Identity per gestire facilmente gli accessi ai servizi di Azure (*Fig. 2*).
È stato configurato un Key Vault, garantendo l'accesso all'identità gestita tramite le policy di accesso e impostando una chiave per l'accesso sicuro alle risorse (*Fig. 3*).
Alla Managed Identity è stato assegnato il ruolo di 'Contributor' nella sottoscrizione per assicurare l'accesso e la gestione di tutte le risorse (*Fig. 4*).

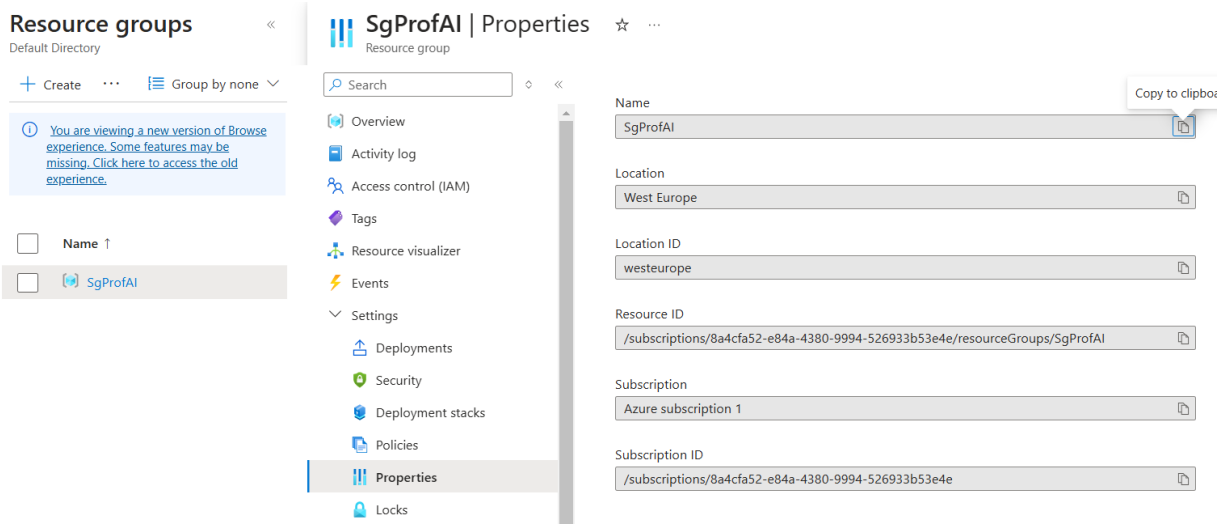


Fig. 1 – Proprietà del Resource Group.

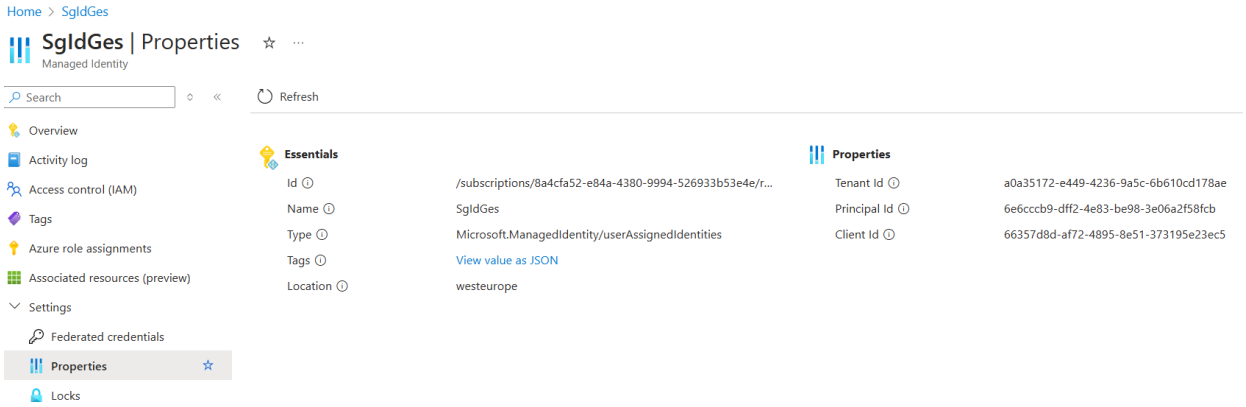


Fig. 2 - Proprietà della Managed Identity.

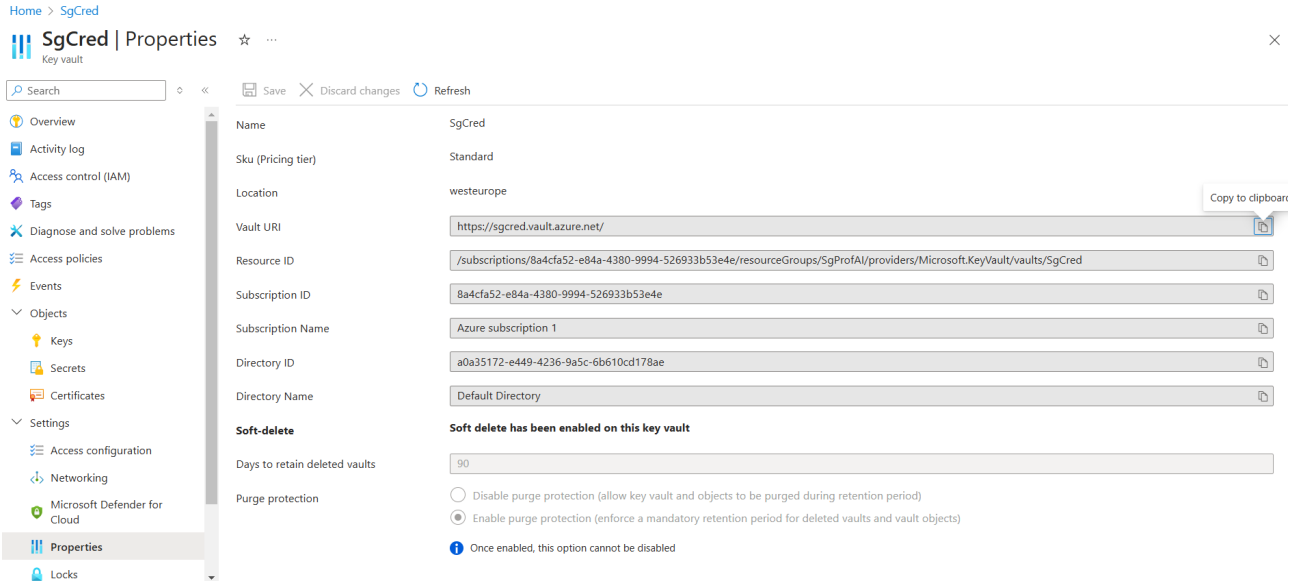


Fig. 3 - Proprietà del Key Vault.

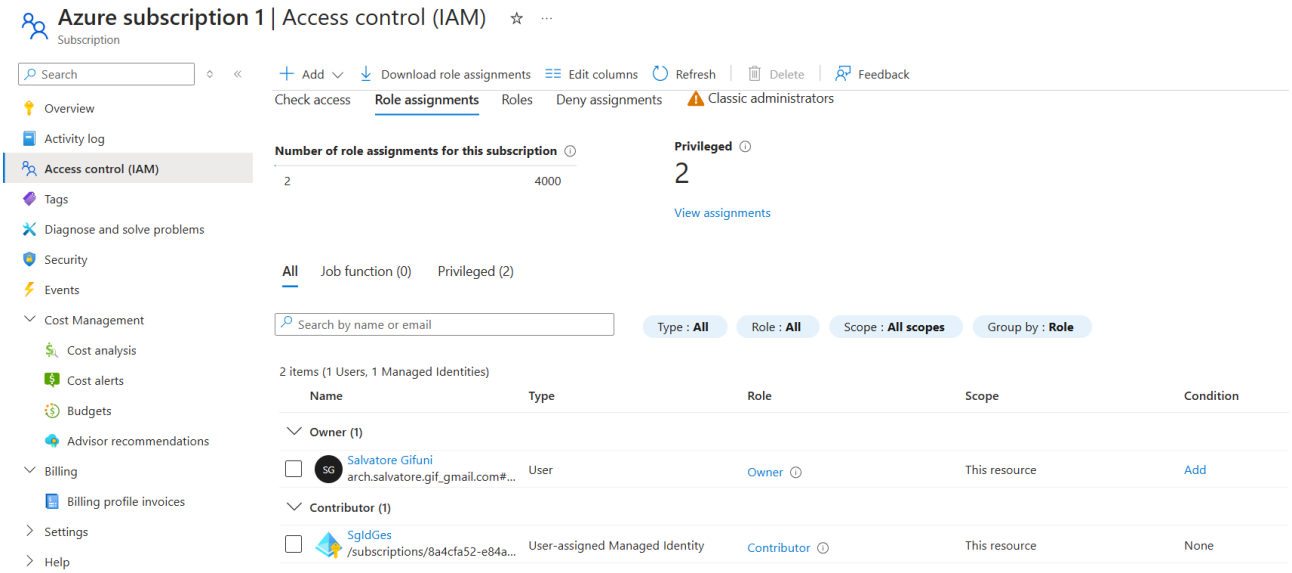
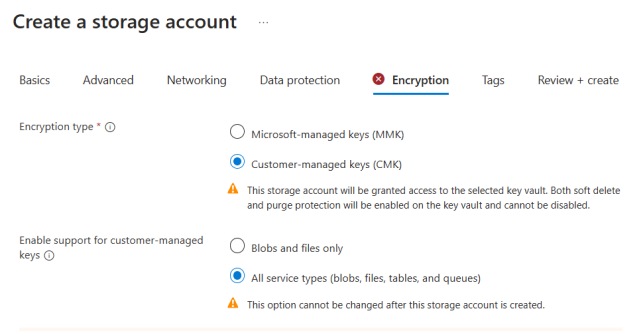


Fig. 4 - Impostazione del ruolo di 'Contributor' alla Managed Identity nella sottoscrizione.

2. Creazione dello *Storage Account*

È stato creato uno Storage Account, ponendo particolare attenzione alle impostazioni di sicurezza nella sezione 'Encryption' (*Fig. 5*).
Sono stati creati tre contenitori: uno per il file di input, uno per i file con modifiche intermedie e uno per il file finale.



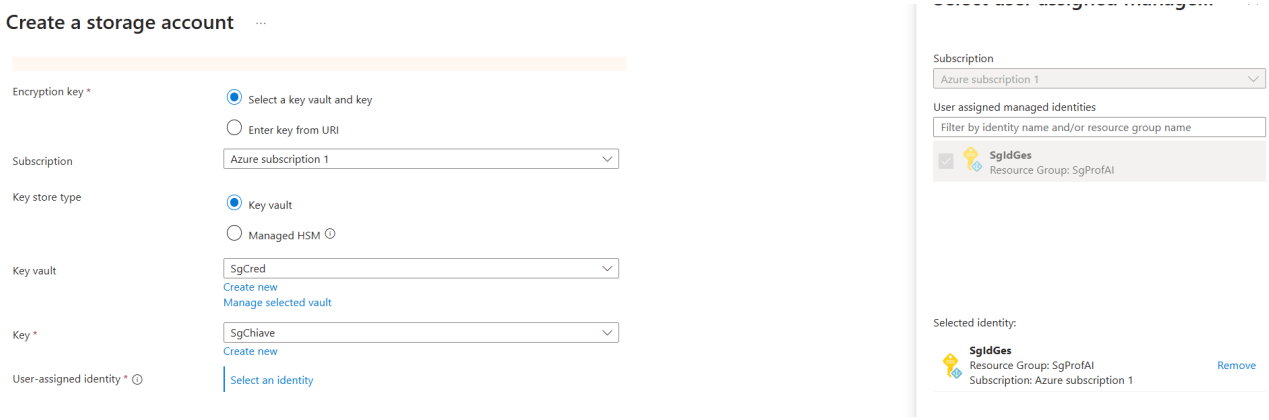


Fig. 5 - Impostazioni 'Encryption' per la creazione dello storage account.

3. Creazione del Translator

È stato creato un Translator, tenendo da parte le chiavi e l'URL per l'utilizzo nell'attività web della pipeline (*Fig. 6*).

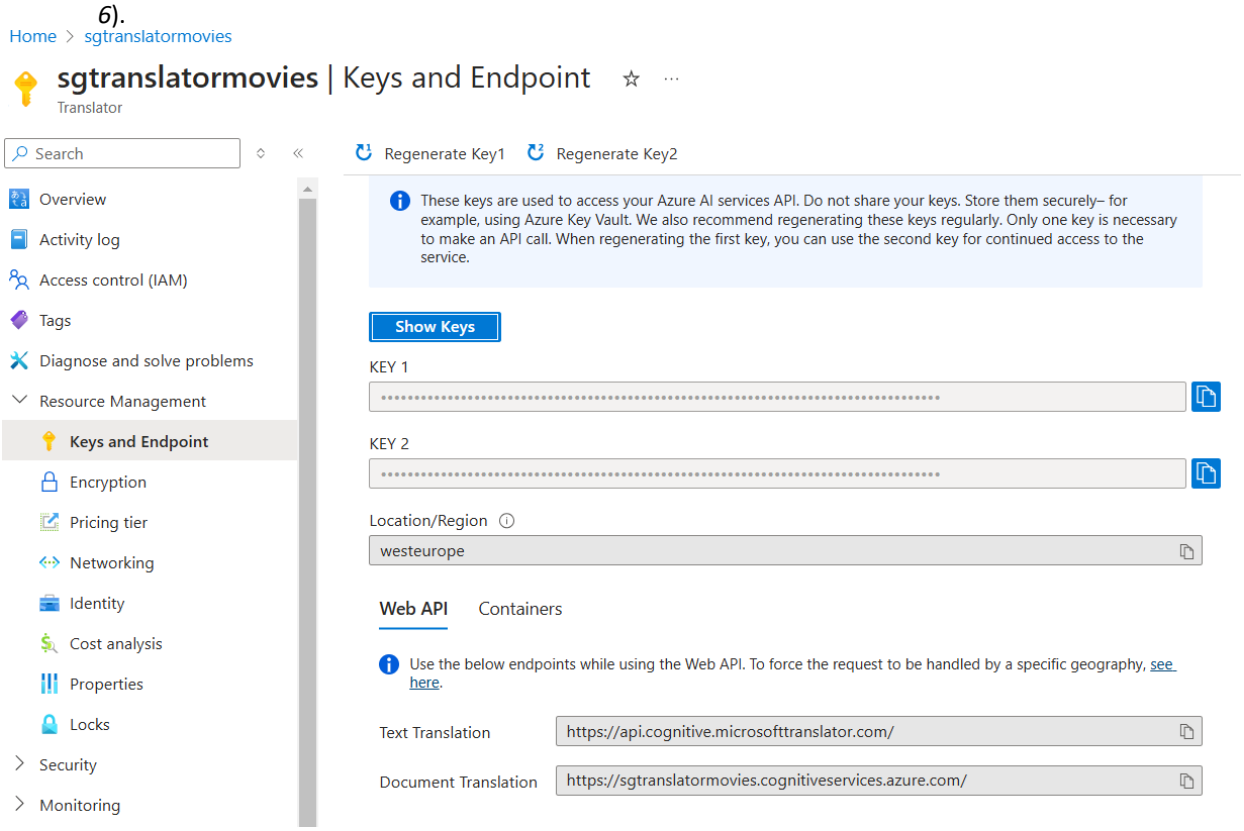


Fig. 6 – Pagina contenente chiavi e URL da passare all'attività Web della pipeline.

4. Creazione del *Data Factory*

È stato creato un Data Factory e nella sezione 'Advanced' (*Fig. 7*) sono state inserite tutte le impostazioni di sicurezza precedentemente configurate.

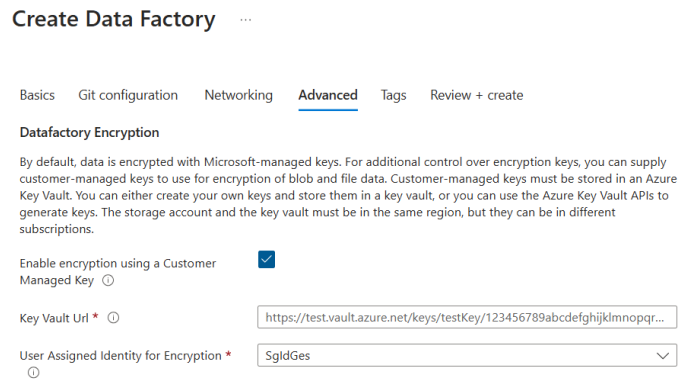


Fig. 7 – Impostazioni di crittografia del Data Factory

5. Creazione della pipeline

La pipeline creata (*Fig. 8*) prende in input, tramite un Data Flow - *'DataWrangling' (Fig. 9*), il dataset originario e seleziona solo le colonne 'Film' [1], 'Generi' e 'Valutazioni'[2], filtra i film mantenendo solo quelli con valutazione superiore a 7[3] (*Fig. 11*) e li ordina in maniera decrescente (opzionale). Il file risultante è salvato in un file intermedio per le attività successive della pipeline.

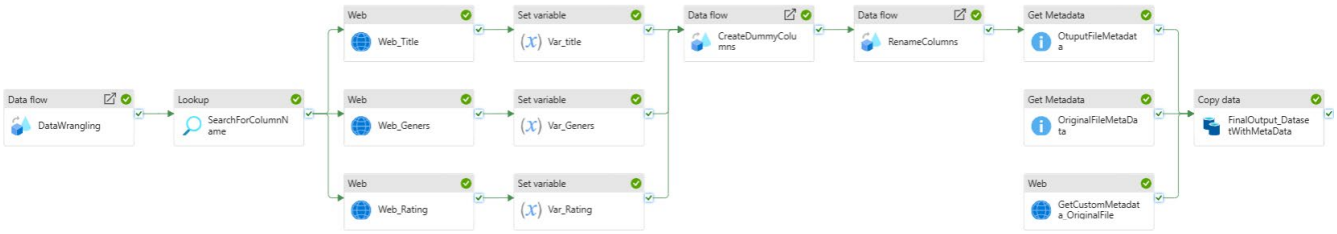


Fig. 8 - Complete Pipeline.

Fig. 9 - Data Flow 'DataWrangling'.

[1] Si è considerata di mantenere la colonna originale 'Movies', ma il procedimento funziona ugualmente con le altre. Inoltre, utilizzando Azure Translator, si nota che la traduzione di "Movies" dall'inglese all'italiano è "Cinema". Successivamente, ho sostituito la traduzione dall'italiano al francese, ottenendo "Film", come richiesto dall'esercizio.

[2] Si sarebbero potuti trasformare i nomi delle colonne direttamente in questo Data Flow al nodo 'SelectCorrectColumnName', ma si è voluto provare ad automatizzare la traduzione dei nomi delle colonne con l'utilizzo successivo di Azure Translator.

[3] I valori della colonna Rating sono stati tramutati in float per consentire il funzionamento del successivo filtro (*Fig. 11)*

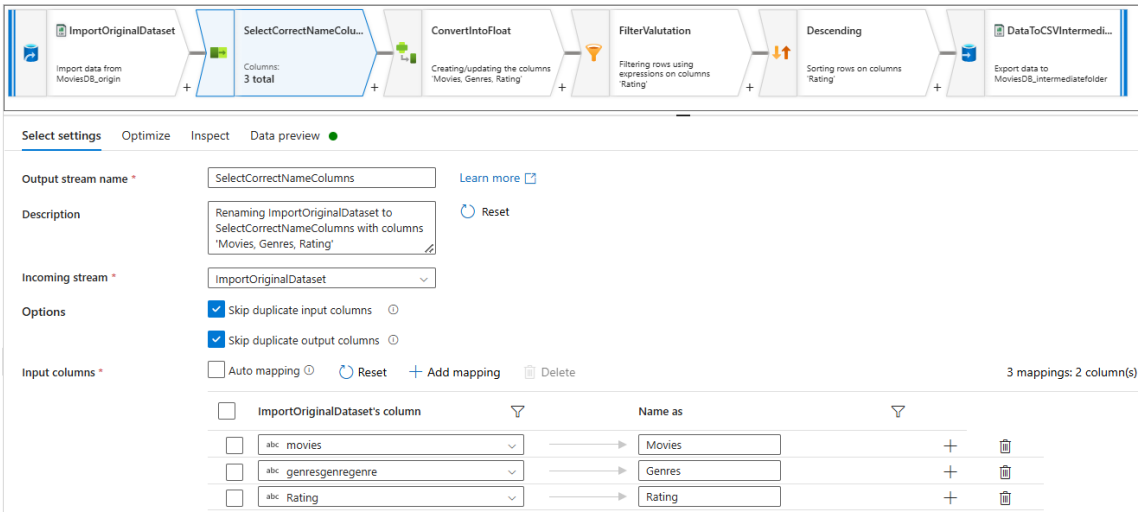


Fig. 10 – Il nodo permette di selezionare solo le 3 colonne richieste, invece delle 5 colonne originali, e viene utilizzato per correggere il formato e gli errori presenti nei nomi così da facilitare le successive attività di traduzione automatica.

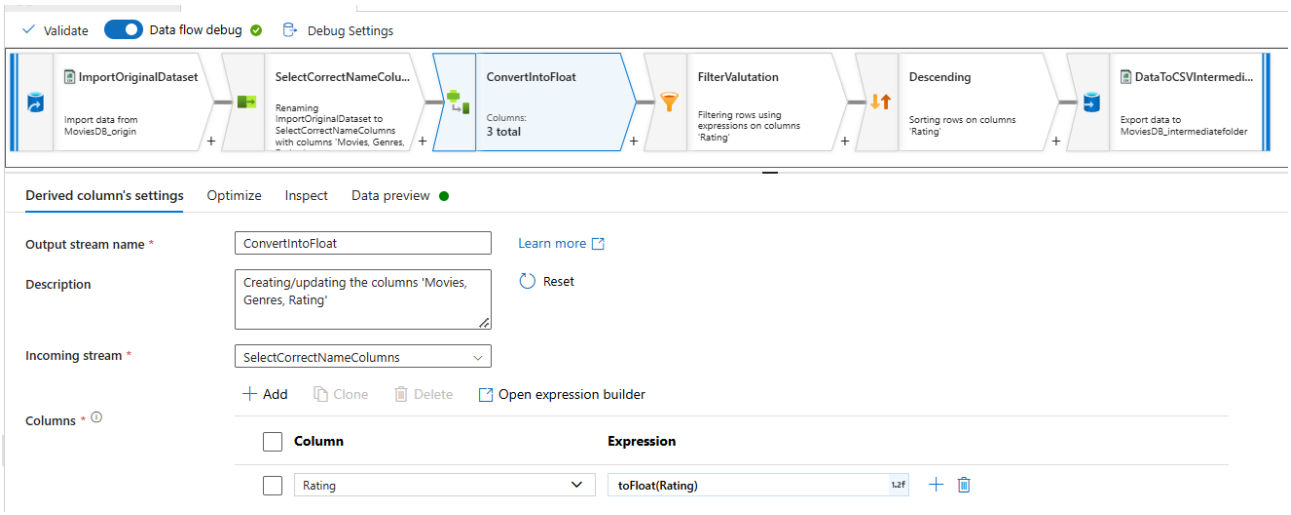


Fig. 11 – Casting dei valori della colonna Rating.

Successivamente, è stato creato un dataset dal file intermedio (*Fig. 12*) disabilitando l'opzione 'first row as header' per consentire all'attività di Lookup – SearchForColumnName di estrapolare i nomi delle colonne. Questi sono stati inviati tramite attività Web (*Fig. 13*) per la traduzione automatica utilizzando Azure Translator (*Fig. 14*).

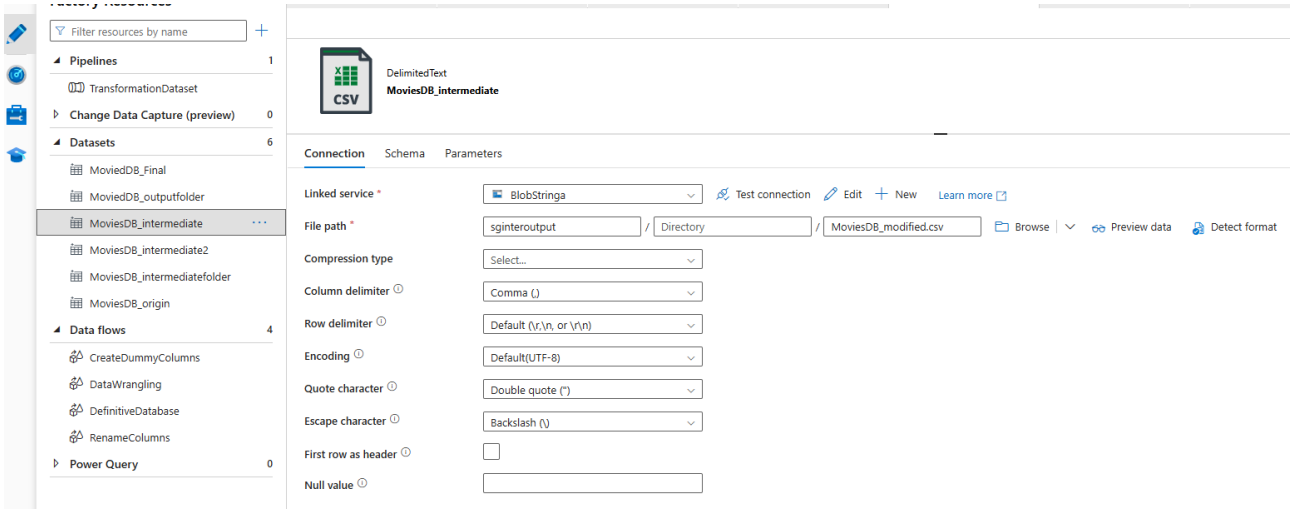


Fig. 12 – Creazione del dataset collegato al file di output dell'attività di 'Data Wrangling'.

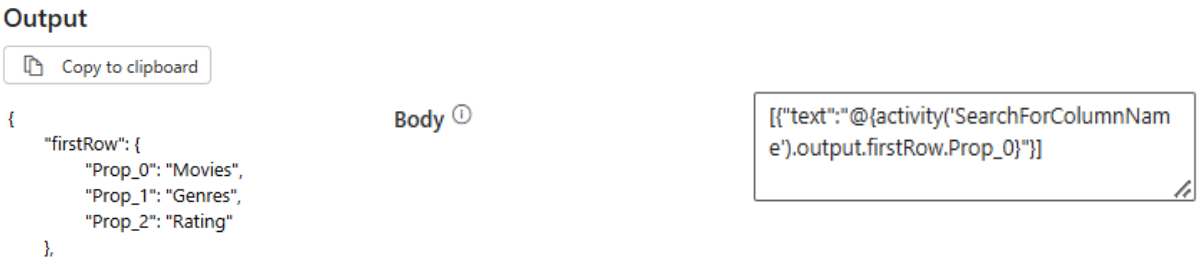


Fig. 13 – A sinistra l'output dell'attività di Look up, a destra il corpo dell'attività Web inviato per la traduzione tramite Azure Translator.

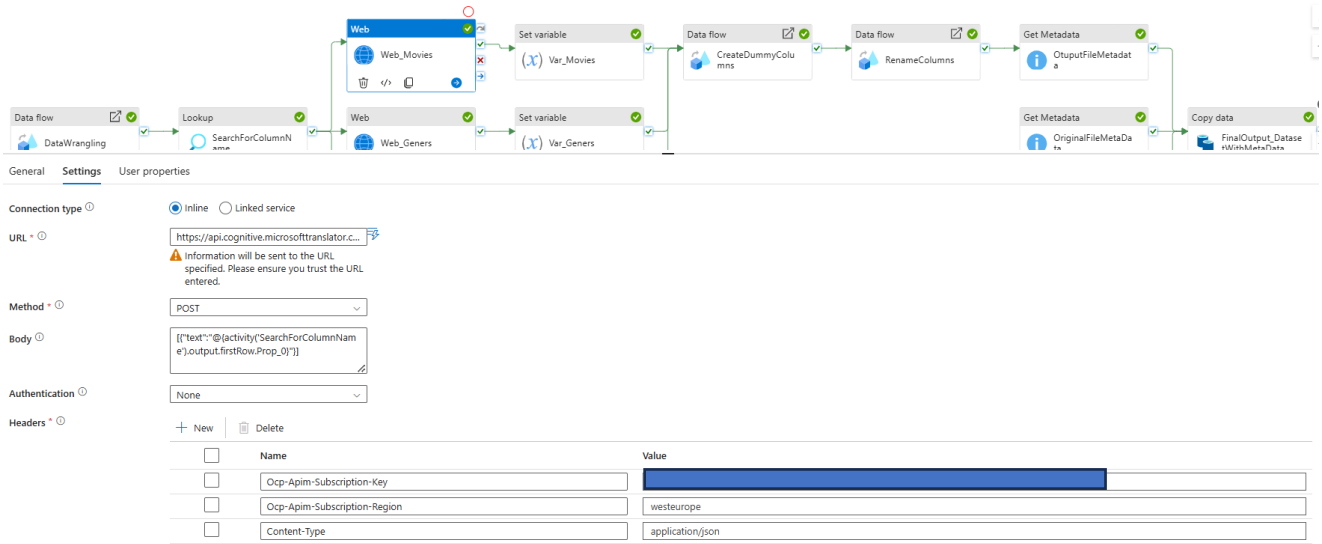


Fig. 14 – Impostazioni dell'attività Web. Nell'URL va inserito oltre al link 'Text Translation' di figura 6, anche 'translate?api version=3.0&from=en&to=it' per specificare la versione dell'API che si sta usando e che si desidera la traduzione del testo dall'inglese all'italiano. Nel primo Headers va riportata la 'key1' della figura 6.

Gli output delle attività Web sono stati salvati come variabili della pipeline, cruciali per le attività successive, infatti, sono stati creati parametri nel Data Flow – RenameColumns (*Fig. 15 e Fig. 16*) per collegare le variabili della pipeline ai nomi delle colonne. (*Fig. 17*).

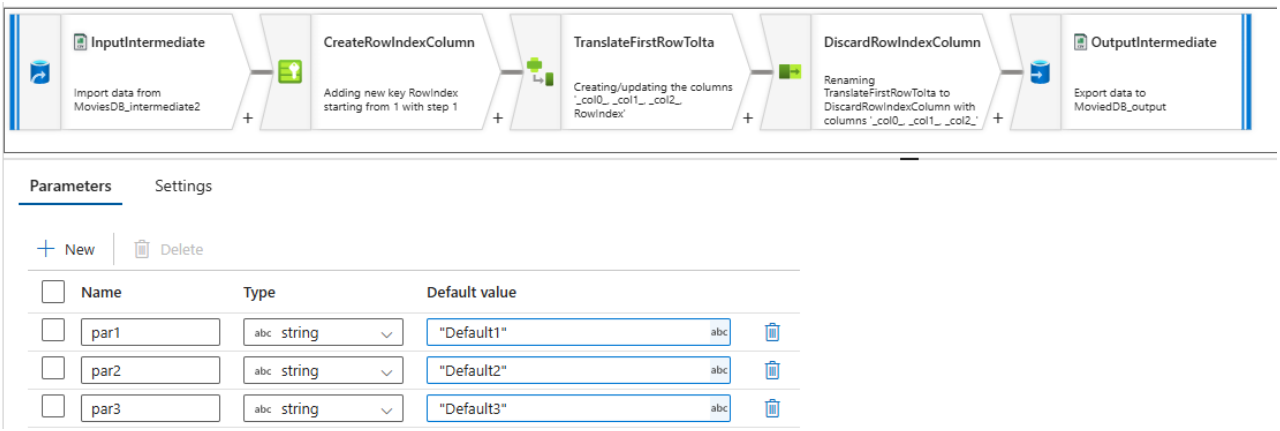


Fig. 15 – Creazione parametri del Data Flow.

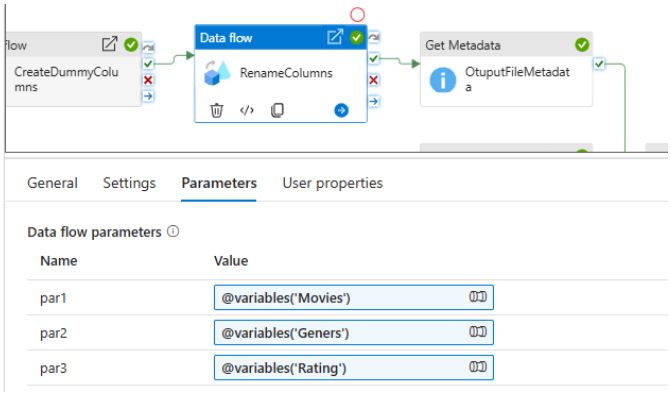


Fig. 16 – Collegamento nella pipeline dei parametri alle variabili del Data Flow.

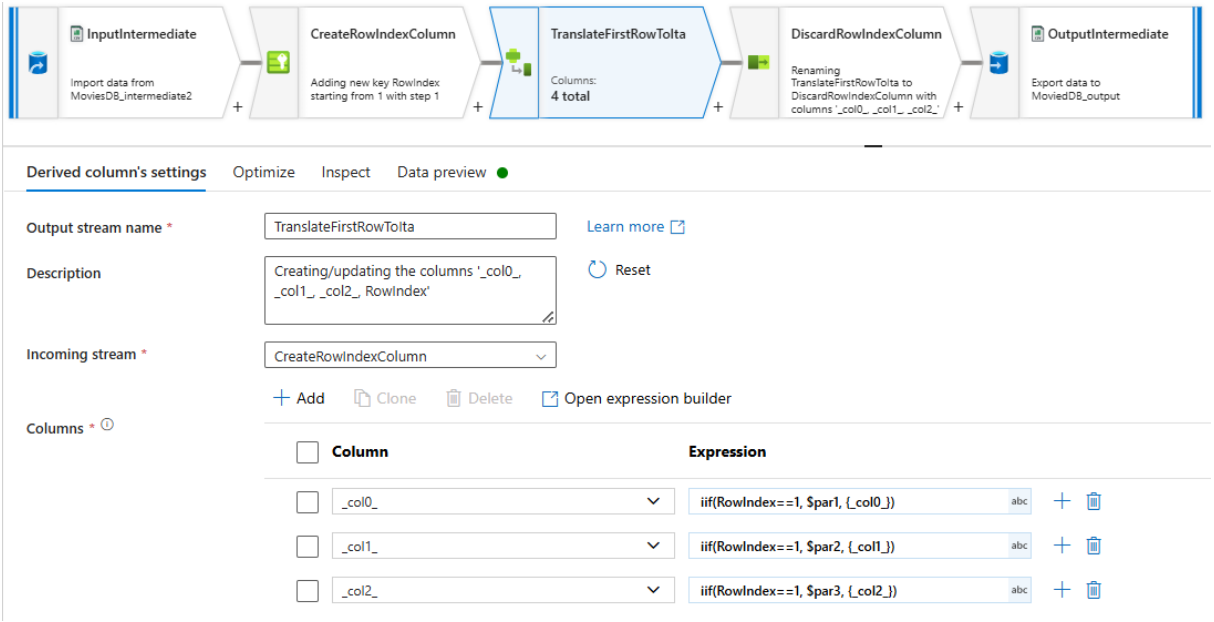


Fig. 17 – Sostituzione delle colonne fittizie con i valori dei parametri del Data Flow, ovvero le variabili della pipeline.

Dato che non c'è attualmente un'attività per modificare i nomi delle colonne dinamicamente, si è trovato l'escamotage di creare delle colonne fittizie utilizzando un ulteriore Data Flow – Create DummyColumns (*Fig. 18*). In questo modo, i nomi delle colonne risultano valori inseriti in una riga del dataset rendodoli sostituibili con i parametri creati tramite un 'Derived Column'.

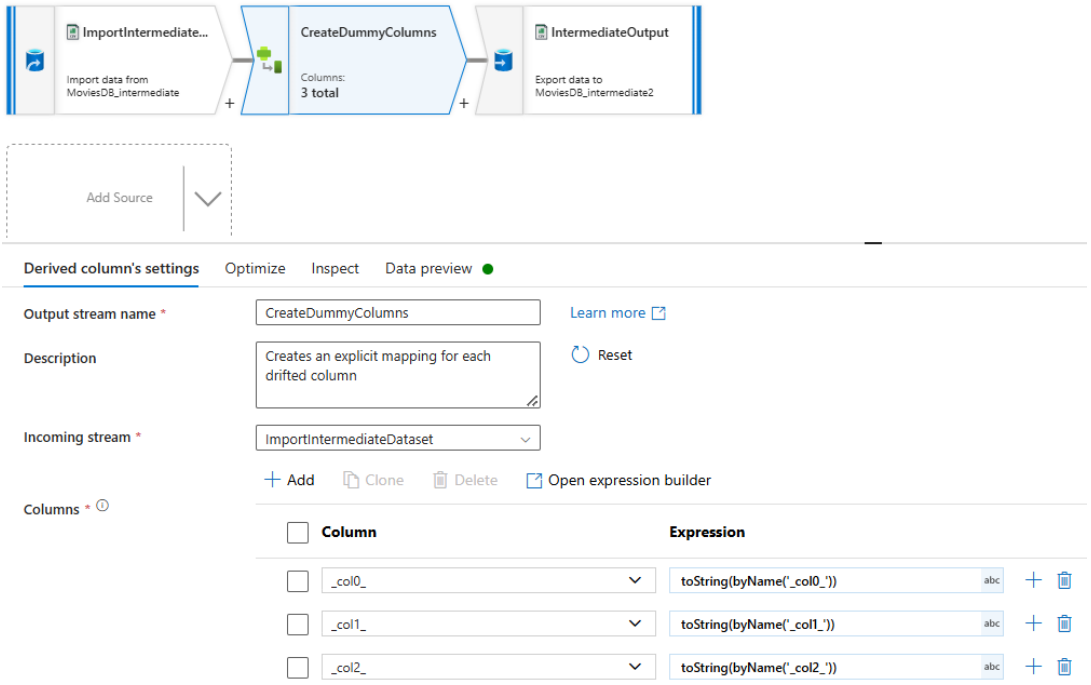


Fig. 18 – Utilizzo di un Data Flow per la creazione di colonne fittizie, per gli scopi descritti.

È stata utilizzata un'attività 'Get Metadata' in parallelo per estrarre informazioni dal dataset originale.

Per ottenere metadati custom (*Fig. 19*), è stata utilizzata un'attività Web collegata tramite una Shared Access Signature (SAS), utilizzando il Blob SAS URL (*Fig. 20*).

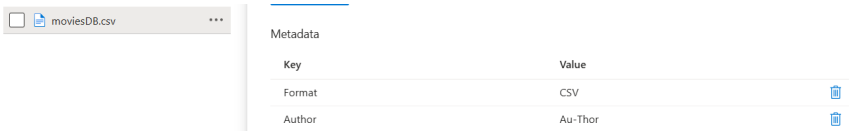


Fig. 19 – Custom Metadata.

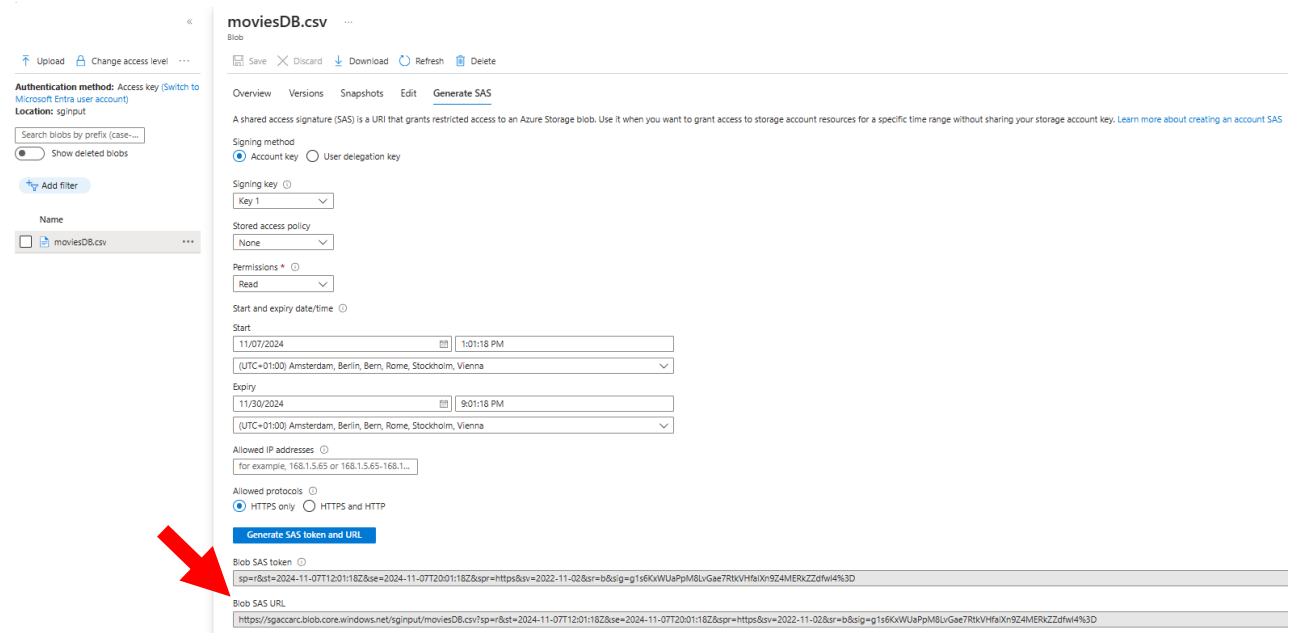


Fig. 20 – URL da inserire nell'attività Web per consentire l'accesso temporaneo al dataset.

Tutti i metadati, sia del dataset originale che del dataset finale, sono stati salvati tramite un'attività di 'Copy data' nel dataset finale di output (Fig.21 e Fig. 22).

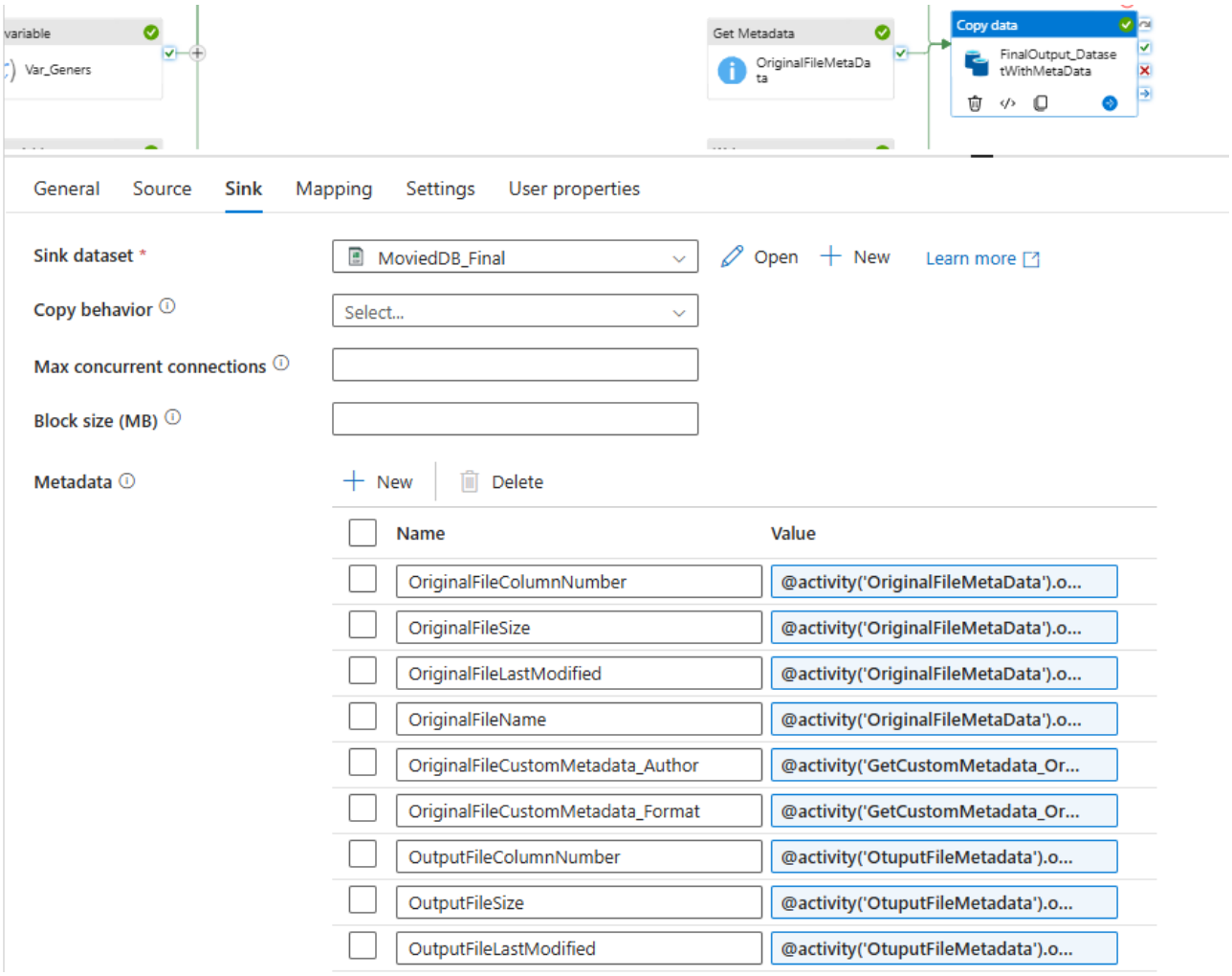


Fig. 21 – Metadata salvati nel file finale di output.

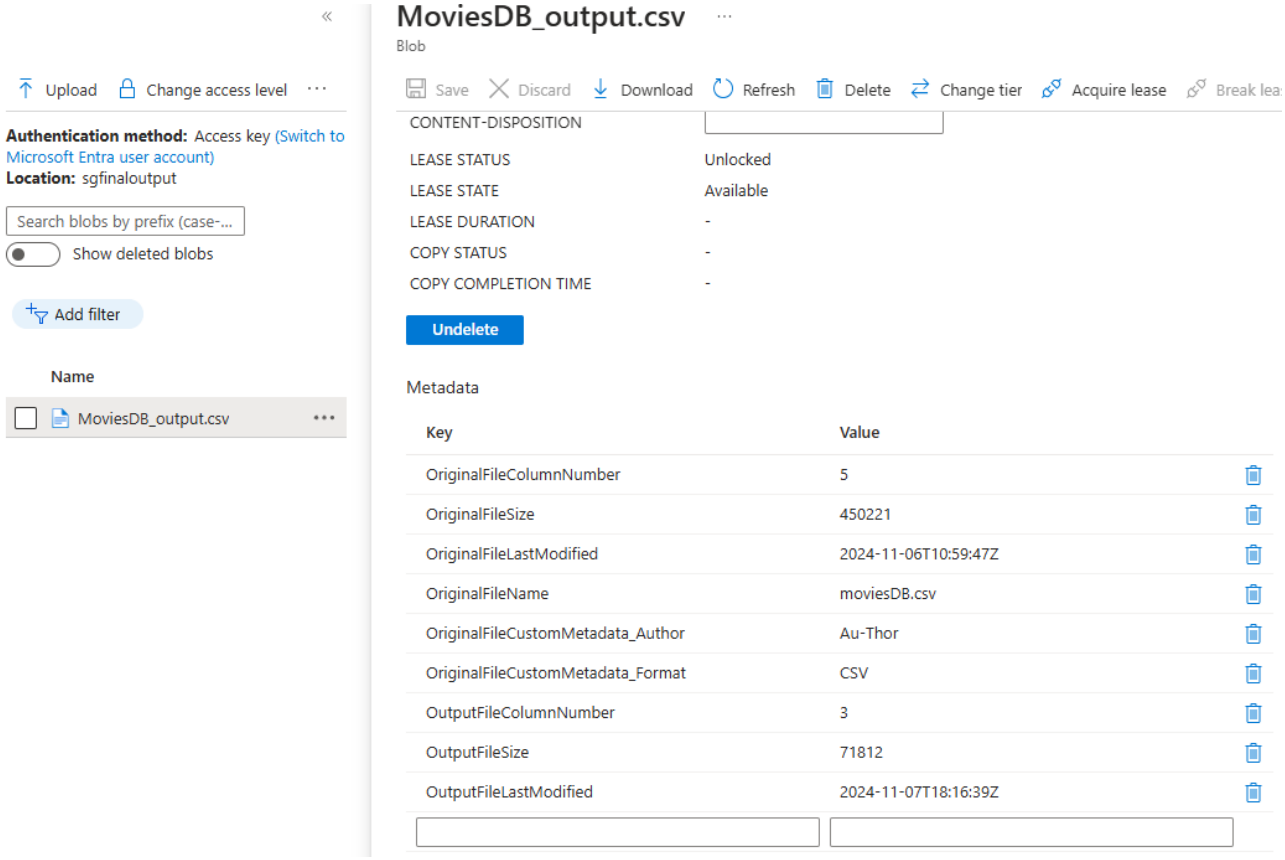


Fig. 22 – Metadata del file di output.