Coda del Workshop del 4 settembre

Step 8

Ripetere tutti i passi sullo schema DWH. Attenzione alle differenze di progettazione!

Step 9

Incapsulare il codice in una stored procedure

Step 10

Aggiungere alla stored procedure un parametro di input per dinamizzare il filtro sul territorio (accettare in input un solo territorio)

Step 11

Aggiungere alla stored procedure un parametro di output che riporta il numero di righe della tabella finale

Step 12

Aggiungere una validazione dell'input. Restituire errore se è inserito un territorio non presente nel Database

Workshop: Creazione di stored procedure per cleaning, caricamento dei dati e log delle modifiche.

Occorre scrivere varie procedure per popolare la tabella *CleanedOrders* (e *CleanedOrders2*) a partire dai dati presenti nella *Orders* (e *Orders2*) e nella *Return*

Step 1

Scrivere una query per individuare tutti le righe presenti nella tabella *Orders* il cui *OrderId*

non è presente nella tabella Return.

Step 2

Scrivere l'istruzione INSERT per inserire le righe dello step1 all'interno della *CleanedOrders*.

Fare attenzione all'elenco e all'ordine delle colonne.

Valorizzare la colonna *DataInizioValidità* con GETDATE(), *DataFineValidità* con NULL e *FlagAttivo* con 1

Step3

Otterremo degli errori di conversione.

Manipolare le colonne contenenti date e numeri per portarli nel formato standard.

Step 4

Se proviamo nuovamente la INSERT. Otterremo un errore di chiave primaria duplicata.

Scrivere la query per individuare le coppie *OrderId* e *ProductID* duplicate.

Per risolvere il problema, a parità di *OrderId* e ProductID, inserire soltanto la riga con *RowID* maggiore.

Suggerimento: aiutiamoci con la **Window Function RANK().** Inseriamo prima la RANK nella SELECT di una CTE (attenzione a cosa mettere nella OVER). Poi eseguiamo il filtro sulla CTE dove la WindowFunction vale 1.

Step 5

Creare una procedura con gestione degli errori e delle transazioni che cancelli le righe esistenti nella *CleanedOrders* e poi esegua la INSERT dei dati.

Step 6

Modificare lo step 5 in modo che le righe cancellate vengano scritte nella tabella *CleanedOrdersLog*.

Step 7

Creare una nuova procedura che, partendo da *Orders*, inserisca le nuove righe nella *CleanedOrders2*, aggiorni quelle precedenti impostando *FlagAttivo* a 0 e la *DataFineValidità* pari alla nuova *DataInizioValidità*

Step 8

Creare una nuova procedura che, partendo dalla tabella *Orders2*, in base alla chiave *OrderId* e *ProductId*, aggiorni le righe già presenti nella tabella *CleanedOrders2* e aggiunga solo quelle nuove.

Step 9

Riscrivere lo step8 inserendo nella tabella *CleanedOrdersLog* la versione precedente delle righe aggiornate (fare attenzione a come valorizzare DataFineValidità).

Step 10

Creare una nuova procedura simile a quella dello step 9, ma dove l'aggiornamento avviene solo per le righe che hanno subito effettivamente delle modifiche.

Step 11

Creare una nuova procedura che, partendo dalla tabella *Orders2,* in base alla chiave *OrderId* e *ProductId*, inserisca sia le nuove righe e sia quelle con delle modifiche rispetto alla *CleanedOrders2*. Inoltre occorre aggiornare le versioni precedenti modificate con *FlagAttivo* a 0 e la *DataFineValidità* pari alla nuova *DataInizioValidità*