UNIVERSITÀ DI PISA

Laboratory of Data Science Report

Federica Di Pasquale (493195)
email: f.dipasquale1@studenti.unipi.it

Ferri Lorenzo (607828)
email: l.ferri11@studenti.unipi.it

Anno accademico 2020/2021

Indice

1 Part 1

1.1 Assignment 0

Il Constellation Schema assegnato è stato riprodotto mediante la query presente nel file "01_DB_creation.sql". Valori nulli sono ammessi solo per gli attributi diversi dalle Chiavi Primarie presenti nelle Dimension Tables. Nelle tre Fact Tables non sono ammessi valori nulli per le Chiavi Esterne, al fine di non perdere ennuple a seguito di uno Star-Join.
I valori della Chiave Primaria della Dimension Table Time, "time_code", sono stati trasformati nel formato yyyy-mm-dd per renderli compatibili con il Data Type "Date" di SQL Server. Sono stati imposti vincoli di Chiave Primaria nelle Dimension Tables e di Chiave Esterna nelle Fact Tables, dunque i file vengono forniti nell'ordine in cui devono essere eseguiti per riempire prima le Dimensions e poi i Facts.

1.2 Assignment 1

Il file "fact.csv" contenente l'intera Fact Table, è stato suddiviso in tre Fact Tables separate, ciascuna per ogni linea di prodotto (cpu, gpu, ram), tramite il file "04_fact.py".
Date le dimensioni complessive della Fact Table, si è scelto di non creare tre file ".csv" ma di suddividere e caricare immediatamente il loro contenuto nel Constellation Schema.
A tale scopo è stato fatto uno scan dell'intero file durante il quale si individua per ogni record la Fact Table corrispondente; si è scelto di non caricare nel DW un record per volta, ma di salvare il loro contenuto in tre liste separate e solo al termine effettuare tre chiamate "executemany(sql, *params)".
Tale scelta si è rivelata molto più efficiente rispetto all'esecuzione di "execute(sql, *params)" effettuata su ogni singolo record; è necessario tuttavia notare che è stato possibile procedere in questo modo solo perché le dimensioni della Fact Table in esame non sono così elevate, mentre in generale sarebbe necessario bilanciare il contenuto caricato in memoria e le chiamate al DB.

1.3 Assignment 2

Le Dimension Tables sono state riempite con i due file: "02_dimensions.py", per tutte le dimensioni tranne Time, e "03_time_dimension.py", per la sola dimensione Time.

- **Dimensions:**
  Una volta effettuata la connessione al DB e definita una query parametrica, sono stati letti i file .csv contenenti le Dimension Tables tramite la funzione "reader" della libreria standard "csv", caricando uno per volta i record così ottenuti.

- **Time Dimension:**
  Sono state preparate le funzioni getQuarter(month) e getDayOfWeek(day, month, year), che ottengono rispettivamente il quarter dato il mese e il giorno della settimana data la

data del giorno. La seconda funzione ha l'obiettivo di calcolare il resto della divisione di x/7, dove x è dato dalla formula:

$$x = year + (year - 1)//4 - (year - 1)//100 + (year - 1)//400 + t \tag{1}$$

questo resto sarà l'indice del giorno della settimana partendo da 'Sabato'. Per ottenere x c'è bisogno di calcolare t, cioè il numero di giorni trascorsi dall'inizio dell'anno. Per farlo si è diviso il calcolo in:

$$t = daysToMonth(month, year) + day \tag{2}$$

dove la funzione daysToMonth(month, year) è una recursione che, dato il mese e l'anno (che serve per verificare se l'anno è bisestile), calcola il numero di giorni trascorsi fino all'inizio del mese corrente sommando i giorni dei mesi a ritroso lungo l'anno.

Il riempimento della tabella Time è avvenuto leggendo il contenuto del file "time.csv" e assegnandolo ad una Dictionary con la funzione DictReader(). Dopo essersi connessi al DB ed aver definito la query parametrica di inserzione, si è iterato sulle righe presenti nella Dictionary per ottenere i valori da passare ai parametri della query che ha riemepito il DB, applicandovi prima le funzioni sopra citate quando necessario.

2 Part 2