

# Importare dati da file CSV all'interno di tabelle PostgreSQL

Gianluca Cima, Maurizio Lenzerini



SAPIENZA  
UNIVERSITÀ DI ROMA

Anno accademico 2021/2022

# Il comando COPY da shell

Per importare file preesistenti in una tabella PostgreSQL si utilizza il comando COPY da usare in una shell del server:

```
COPY table_name FROM file_name [ [WITH] (option [, ...])]
```

dove **option** può avere:

- *formato\_file*
- DELIMITER AS '*delimiter\_character*'
- NULL AS '*null\_string*'
- HEADER

Per maggiori dettagli di veda la documentazione ufficiale:

<https://www.postgresql.org/docs/9.6/sql-copy.html>

# Il comando IMPORT/EXPORT in PgAdmin

Per importare o esportare file preesistenti in una tabella PostgreSQL si può utilizzare anche il comando IMPORT/EXPORT in PgAdmin.

Per l'importazione, una volta creata la tabella con la struttura che accoglierà il file, occorre azionare il tasto destro del mouse sopra il nome della tabella, e selezionare il comando IMPORT. A fronte di questa scelta si aprirà un menu mediante il quale si potranno scegliere i parametri che abbiamo discusso nella slide precedente.

# Esempio: Open Data Ancitel - Parte 1

- Consideriamo il dataset contenente le informazioni dei dati territoriali e demografici dei comuni italiani, preso dal portale Open Data di Ancitel e per comodità reperibile nella sezione della pagina MOODLE del corso dedicata alla esercitazione 6
- Impostiamo i permessi di lettura da fonti esterne per questo file:
  - Windows: Tasto destro sul file, Proprietà, tab Sicurezza, Modifica, Aggiungi, Scrivere *Everyone* nel box “Immettere il nome dell’oggetto da selezionare”, Ok, Applica;
  - Linux: Aprire una shell, digitare “`chmod a+x nome_file`”.

Nella prossima slide vedremo come importare questo file CSV all’interno di una tabella PostgreSQL mediante il comando COPY.

## Esempio: Open Data Ancitel - Parte 2

- Creiamo la tabella che conterrà il contenuto del file CSV:

```
create table comune(  
Comune varchar, ISTAT varchar, Provincia varchar,  
SiglaProv varchar, Regione varchar, AreaGeo varchar,  
PopResidente integer, PopStraniera integer,  
DensitaDemografica varchar, SuperficieKmq varchar,  
AltezzaCentro integer, AltezzaMinima integer,  
AltezzaMassima integer, ZonaAltimetrica varchar,  
TipoComune varchar, GradoUrbaniz varchar,  
IndiceMontanita varchar, ZonaClimatica varchar,  
ZonaSismica varchar, ClasseComune varchar,  
Latitudine varchar, Longitudine varchar);
```

- Importiamo ora il contenuto del file CSV all'interno della tabella comune appena creata:

```
COPY comune FROM 'C:\\Users\\Gianluca\\Desktop\\comuni.csv'  
WITH CSV DELIMITER AS ';' HEADER;
```

# Importazione di file CSV su PostgreSQL

## 1) Scaricare i file csv

Accendere la macchina virtuale, connettersi alla pagina moodle del corso e scaricare i file in formato “.csv”. Mettere i file “.csv” appena scaricati in una nuova cartella chiamata “Esercitazione” sul Desktop della macchina virtuale.

## 2) Aprire un terminale Linux

Lanciare un terminale “Linux” dalla macchina virtuale premendo sull’apposita icona nella barra in basso (vedi immagine seguente)



## 3) Spostarsi nella cartella “Esercitazione” sul Desktop

Eseguire il comando:

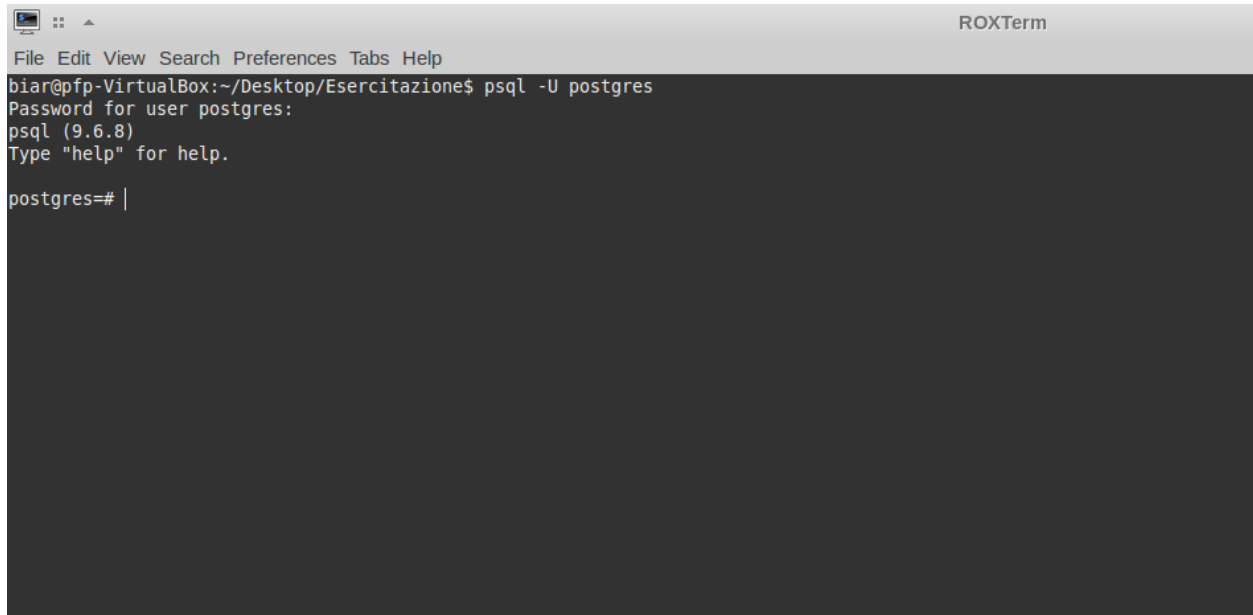
```
$cd /home/biar/Desktop/Esercitazione/
```

## 4) Connettersi a PostgreSQL usando la shell

Eseguire il comando:

```
$psql -U postgres
```

Il sistema chiederà di inserire una password. Digitare “biar”



```
File Edit View Search Preferences Tabs Help
biar@pfp-VirtualBox:~/Desktop/Esercitazione$ psql -U postgres
Password for user postgres:
psql (9.6.8)
Type "help" for help.

postgres=# |
```

## 5) Importare i file csv

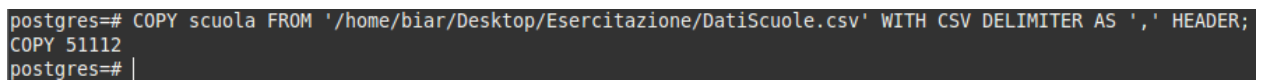
Eeguire il seguente comando:

```
postgres=# COPY <nometabella> '<path/completo/al/file/csv>' WITH CSV DELIMITER AS ',' HEADER;
```

**Attenzione:** La tabella <nometabella> deve essere già presente in PostgreSQL prima di invocare il comando, e deve avere un numero di attributi pari al numero di valori in ogni riga del file csv.

Ad esempio, nel caso della tabella “Scuola”, per importare i dati contenuti nel file “DatiScuole.csv”, eseguire il seguente comando:

```
postgres=# COPY scuola FROM '/home/biar/Desktop/Esercitazione/DatiScuole.csv' WITH CSV DELIMITER
AS ',' HEADER;
```



```
postgres=# COPY scuola FROM '/home/biar/Desktop/Esercitazione/DatiScuole.csv' WITH CSV DELIMITER AS ',' HEADER;
COPY 51112
postgres=# |
```

Corso di Basi di dati A.A. 2021/22  
Esercitazione 6  
applicazione che utilizza stored procedure, trigger e JDBC

Gianluca Cima, Federico Croce, Maurizio Lenzerini

## 1. Il contesto generale

La nostra azienda vuole realizzare un'applicazione per il Ministero dell'Istruzione, dell'Università e della Ricerca che consenta di gestire l'anagrafe degli istituti scolastici e delle scuole e di registrare le valutazioni di tali istituti, associandole alle scuole. Ricordiamo che un istituto scolastico (detto anche "plesso") è una struttura che comprende una o più scuole. Sono gli istituti scolastici ad essere valutati secondo diversi criteri.

Il nostro manager ha già individuato due "open data sets", ovvero due file .csv disponibili in rete con dati rilevanti per la nostra applicazione. I due file sono:

- il file "DatiScuole.csv", contenente i dati relativi alle varie scuole nel territorio italiano e gli istituti scolastici ai quali le scuole appartengono,
- il file "DatiValutazioniIstituti.csv", contenente i dati relativi ad alcune valutazioni degli istituti per l'anno scolastico 2016-2017, in termini di un voto da 1 a 7 assegnato agli istituti relativamente ai diversi criteri. Si può assumere che tutti gli istituti che compaiono nel file "DatiValutazioniIstituti.csv" compaiano anche nel campo giusto del file "DatiScuole.csv".

Abbiamo già preparato i due file per l'esercitazione, che si possono scaricare dalla pagina MOODLE del corso, nella sezione relativa alla esercitazione 6.

## 2. Cosa prevede l'esercitazione

L'esercitazione consiste nell'eseguire i seguenti compiti assegnatici dal nostro manager:

1. Definire una base di dati PostgreSQL con due tabelle, ciascuna per memorizzare i dati contenuti in uno dei due file. Per definire correttamente le tabelle occorre analizzare il contenuto dei due file, in particolare l'intestazione, che ci dice quanti e quali campi dobbiamo prevedere per le tabelle. Per popolare correttamente le tabelle occorre importare i dati dai file csv.
2. Arricchire la tabella relativa alle scuole definendo opportuni attributi per rappresentare, per ogni scuola
  - a. il numero di valutazioni ricevute dall'istituto di quella scuola; *senza considerare i criteri*
  - b. la somma numerica dei voti delle valutazioni ricevute dall'istituto di quella scuola;
  - c. la media aritmetica dei voti delle valutazioni ricevute dall'istituto di quella scuola (se la media non è definita il valore deve essere NULL).

Si noti che il manager ci chiede di assegnare ad ogni scuola i dati relativi alle valutazioni dell'istituto a cui la scuola appartiene. Se anche questa richiesta può sembrare strana, noi dobbiamo fare quello che ci chiede il manager.



3. Definire un insieme di istruzioni SQL per inserire nella tabella relativa alle scuole i dati corretti nei nuovi attributi definiti rispondendo al punto precedente.
4. Creare opportuni trigger che consentano di tenere aggiornati, ad ogni eventuale modifica della tabella relativa alle valutazioni, i tre attributi aggiuntivi creati al passo precedente. Le possibili modifiche corrispondono, ovviamente, alle operazioni di insert, delete, update sulle valutazioni nell'anno scolastico 2016-2017. Si tenga conto che nell'aggiornamento di una valutazione è possibile cambiare solo il criterio ed il punteggio.
5. Scrivere un programma JAVA che realizza la funzionalità di supporto che l'applicazione vuole dare al ministero. In particolare, il programma deve basarsi su un ciclo in cui si chiede all'utente di specificare quale tra 4 possibili comandi vuole eseguire, immettendo un valore intero compreso tra 1 a 4. Qui di seguito indichiamo cosa deve fare il programma a fronte del valore scelto dell'utente:

- *Valore 1: terminazione del programma.* Se l'utente sceglie questa opzione, occorre terminare il programma, dopo aver mostrato all'utente i primi 10 istituti con la media più alta dei voti delle valutazioni (mostrando anche la media).
- *Valore 2: riassunto dei dati sulle regioni.* Per ogni regione per la quale il numero N degli istituti che hanno ricevuto valutazioni per il solo criterio 21 è maggiore di 0, occorre mostrare all'utente la regione ed il numero N.
- *Valore 3: inserimento/aggiornamento di una valutazione di un istituto.* All'utente che scelga questa opzione occorre chiedere il codice dell'istituto a cui si riferisce la nuova valutazione (sempre l'anno scolastico 2016-2017), il criterio di valutazione ed il voto relativo ed inserire o aggiornare la relativa valutazione nella tabella opportuna. In particolare, se una valutazione per l'istituto ed il criterio indicato dall'utente già esiste, allora occorre eseguire l'aggiornamento del voto con il nuovo valore. Se non esiste una valutazione per l'istituto ed il criterio indicato dall'utente e l'istituto compare già nella tabella delle scuole, allora occorre inserire il record corrispondente alla nuova valutazione. Se una valutazione per l'istituto ed il criterio indicato dall'utente non esiste e l'istituto non compare nella tabella delle scuole, allora occorre rigettare l'operazione.
- *Valore 4: cancellazione di una valutazione.* All'utente che scelga questa opzione occorre chiedere un criterio di valutazione ed una provincia, e bisogna eliminare tutte le valutazioni di istituti di quella provincia riguardanti quel criterio.

Si noti che il programma deve effettuare tutte le operazioni suddette in modo da garantire la correttezza della base di dati e in modo che dal programma si esca solo quando l'utente sceglie l'opzione 1.

### 3. Panoramica del software da realizzare

Il sistema software da realizzare è quindi costituito dalla classica architettura a tre livelli, con

1. Presentation layer
2. Application layer
3. Data layer

Il presentation layer è formato dalla interfaccia a menù che il programma JAVA presenta all'utente per fare scegliere all'utente stesso le operazioni da svolgere (fino all'uscita dal programma).

L'application layer è costituito dai 4 servizi che il programma JAVA offre per essere attivati a fronte delle possibili 4 scelte dell'utente.

Il data layer è costituito dalla base di dati gestita dal DBMS PostgreSQL, con tutte le sue componenti (comprese stored procedure e trigger).