

# pgloader

## PG DAY UK

Dimitri Fontaine  
@tapoueh

July 9th, 2014

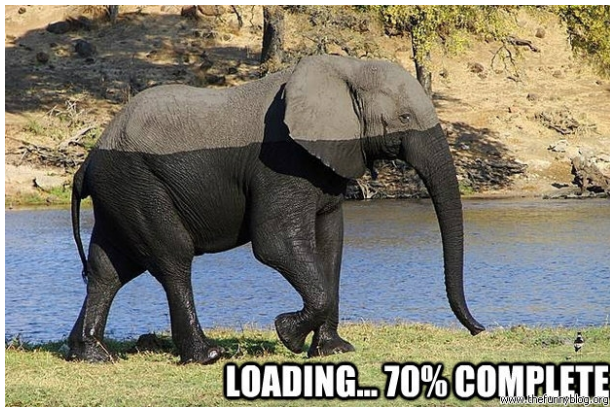
# Load data into PostgreSQL. Fast.

`http://pgloader.io/`



# Load data into PostgreSQL. Fast.

<http://pgloader.io/>



## Load from CSV files

`http://pgloader.io/howto/csv.html`



## Load from dBase III

`http://pgloader.io/howto/dBase.html`



## Load from SQLite

<http://pgloader.io/howto/sqlite.html>



# Load from a live MySQL connection

`http://pgloader.io/howto/mysql.html`



# Load from a live MySQL connection

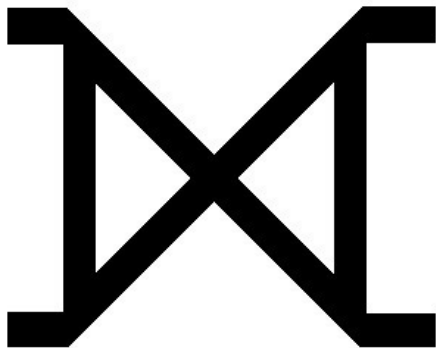
`http://pgloader.io/howto/mysql.html`





# pgloader : Transform data on the fly

`http://pgloader.io/`

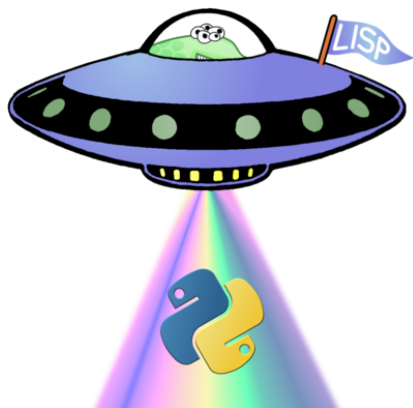


# Load data into PostgreSQL. Fast.

`http://pgloader.io/`



Used to be *python* code. Now Common Lisp.



## pgloader got faster

```
select rows, v2, v3,  
       round(( extract(epoch from v2)  
             / extract(epoch from v3))::numeric, 2)  
       as speedup  
from timing;
```

rows		v2		v3		speedup
4768765	@ 37 mins	10.878	@ 1 min	26.917		25.67
3115880	@ 36 mins	5.881	@ 1 min	10.994		30.51
3865750	@ 33 mins	40.233	@ 1 min	15.33		26.82
3994483	@ 29 mins	30.028	@ 1 min	18.484		22.55

(4 rows)

## Now using a command language

LOAD CSV

FROM inline (x, y, a, b, c, d)

INTO postgresql:///pgloader?csv (a, b, d, c)

WITH truncate,

skip header = 1,

fields optionally enclosed by '"',

fields escaped by double-quote,

fields terminated by ','

SET client\_encoding to 'latin1',

work\_mem to '12MB',

standard\_conforming\_strings to 'on'

## with extra before/after sections

```
BEFORE LOAD DO
  $$ drop table if exists csv; $$,
  $$ create table csv (
    a bigint,
    b bigint,
    c char(2),
    d text
  );
  $$;
```

## Some data source examples

```
FROM stdin
```

```
FROM inline (a, b, c)
```

```
FROM data/2013_Gaz_113CDs_national.txt
```

```
FROM FILENAME MATCHING ~/GeoLiteCity-Location.csv/
```

```
FROM ALL FILENAMES MATCHING ~/ALIOR/
```

```
FROM ALL FILENAMES MATCHING ~/F[A-Z]{4}1[45]|OZ20/
```

```
FROM http://www.census.gov/geo/maps-data/  
data/docs/gazetteer/places2k.zip
```

```
FROM http://www.insee.fr/fr/methodes/nomenclatures/  
cog/telechargement/2013/dbf/historiq2013.zip
```

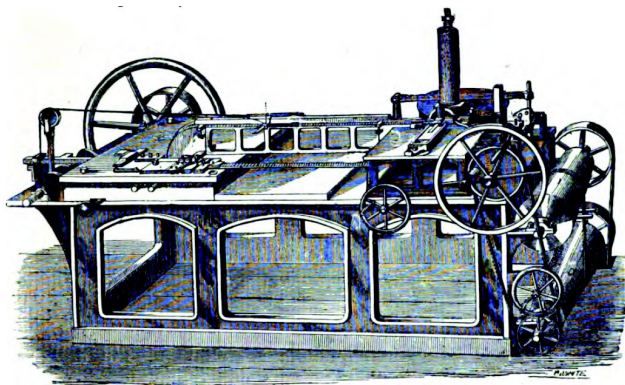
## Transformation de données

```
FROM FILENAME MATCHING ~/GeoLiteCity-Blocks.csv/  
  WITH ENCODING iso-8859-1  
  (  
    startIpNum, endIpNum, locId  
  )  
INTO postgresql:///ip4r?geolite.blocks  
  (  
    iprange ip4r using (ip-range startIpNum endIpNum),  
    locId  
  )
```



# MySQL et les types de données

Text vide ou NULL, valeurs par défaut, dates à 0000-00-00, int(11), float(20,2), tinyint mais pas de boolean, sets, encodage, ...



JOHNSON AND ATKINSON'S TYPE CASTER.

Digitized by Google

# Load data into PostgreSQL. Fast.

`http://pgloader.io/`

