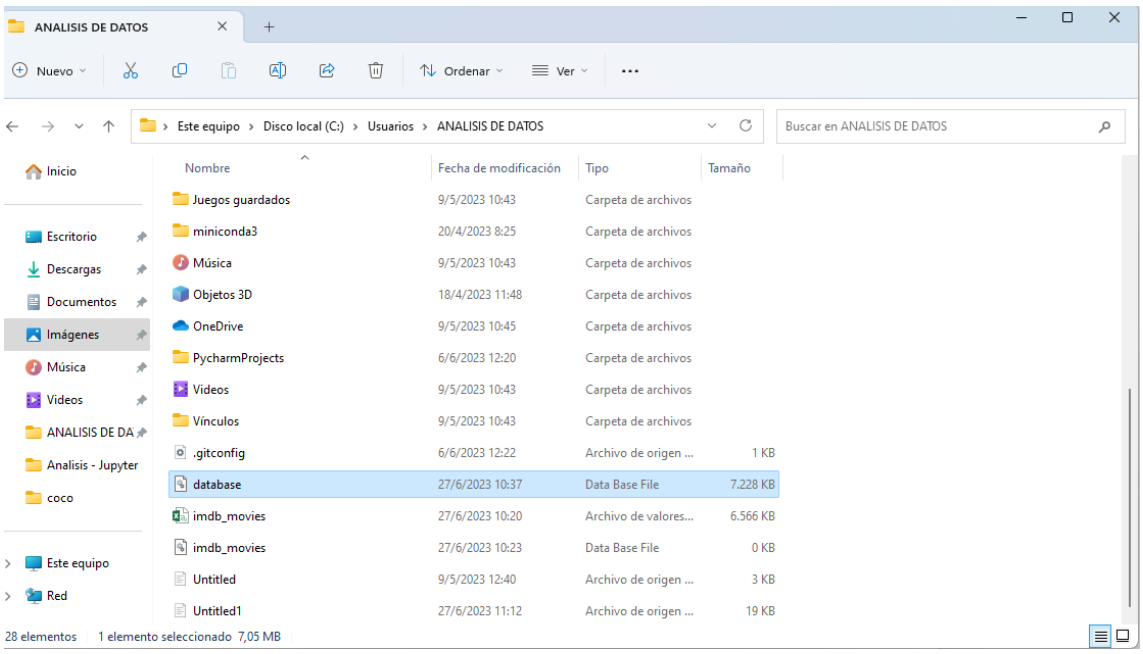
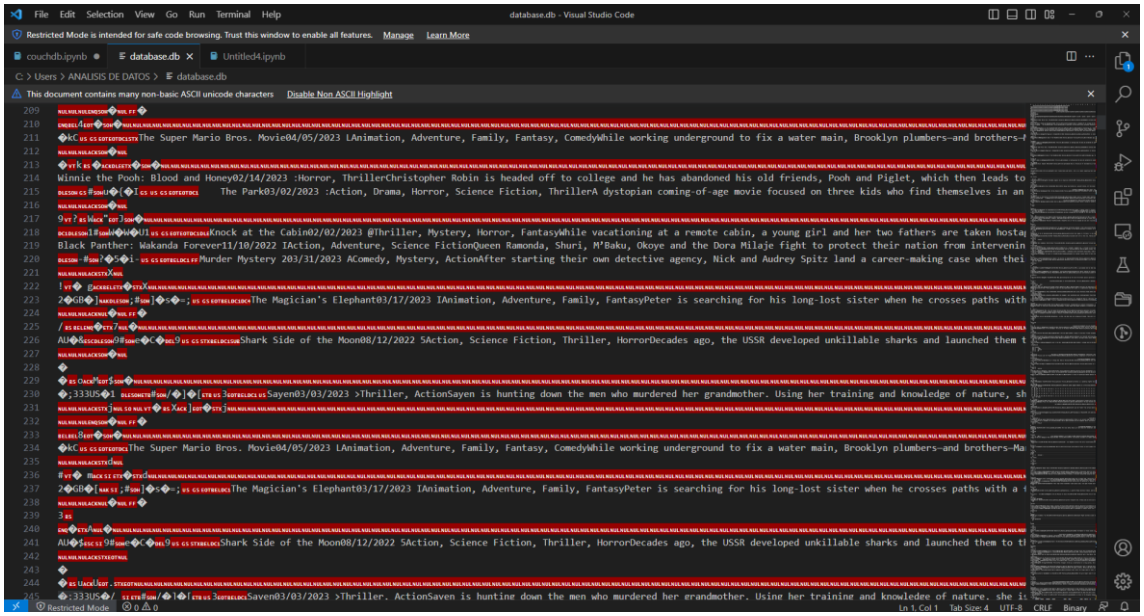
Nombre: Danny Yanacallo

1.- Importar el csv a SQLite (2 puntos)

```
In [1]: pip install sqlite3

Note: you may need to restart the kernel to use updated packages.

ERROR: Could not find a version that satisfies the requirement sqlite3 (from versions: none)
ERROR: No matching distribution found for sqlite3

In [38]: import sqlite3
         import pandas as pd

In [13]: con=sqlite3.connect("database.db")

In [14]: df=pd.read_csv("imdb_movies.csv")

In [15]: df.to_sql("Peliculas", con, if_exists="replace", index=False)

Out[15]: 10178

In [18]: df

Out[18]:
```

	names	date_x	score	genre	overview	crew	orig_title	status	orig_lang	budget_x	r
0	Creed III	03/02/2023	73.0	Drama, Action	After dominating the boxing world, Adonis Cree...	Michael B. Jordan, Adonis Creed, Tessa Thompso...	Creed III	Released	English	75000000.0	2.7161
1	Avatar: The Way of Water	12/15/2022	78.0	Science Fiction, Adventure, Action	Set more than a decade after the events of the...	Sam Worthington, Jake Sully, Zoe Saldaña, Neyt...	Avatar: The Way of Water	Released	English	460000000.0	2.3167
2	The Super Mario Bros	04/05/2023	76.0	Animation, Adventure, Family, Fantasy, Comedy	While working underground	Chris Pratt, Mario (voice), Anya	The Super Mario Bros	Released	English	100000000.0	7.2445





Codigo:

#Importar librerias

pip install sqlite3
import sqlite3
import pandas as pd

#Conectarnos a la base de datos
con=sqlite3.connect("database.db")
#leer la base de datos
df=pd.read_csv("imdb_movies.csv")
# Importar el csv a SQLite
df.to_sql("Peliculas", con, if_exists="replace", index=False)
df

2.- Transformación de cada elemento del csv en json y carga en couchdb mediante código. (8 puntos)

In [20]:
```
pip install couchdb
```

```
Collecting couchdb
  Using cached CouchDB-1.2-py2.py3-none-any.whl (67 kB)
Installing collected packages: couchdb
Successfully installed couchdb-1.2
Note: you may need to restart the kernel to use updated packages.
```

In [21]:
```
import couchdb
```

In [22]:
```
couch=couchdb.Server('http://admin:admin@127.0.0.1:5984/')
```

In [24]:
```
db=couch.create('peliculas')
```

In [25]:
```
db=couch['peliculas']
```

In [26]:
```
pip install csv
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
ERROR: Could not find a version that satisfies the requirement csv (from versions: none)
ERROR: No matching distribution found for csv
```

In [27]:
```
pip install json
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
ERROR: Could not find a version that satisfies the requirement json (from versions: none)
ERROR: No matching distribution found for json
```

In [27]:
```
pip install json
```

```
Note: you may need to restart the kernel to use updated packages.
```

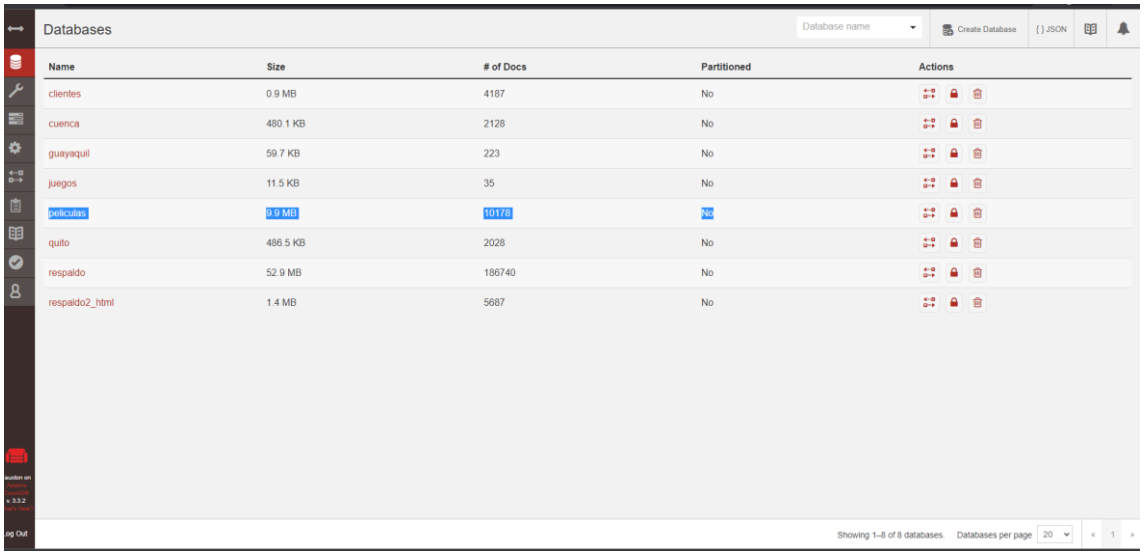
```
ERROR: Could not find a version that satisfies the requirement json (from versions: none)
ERROR: No matching distribution found for json
```

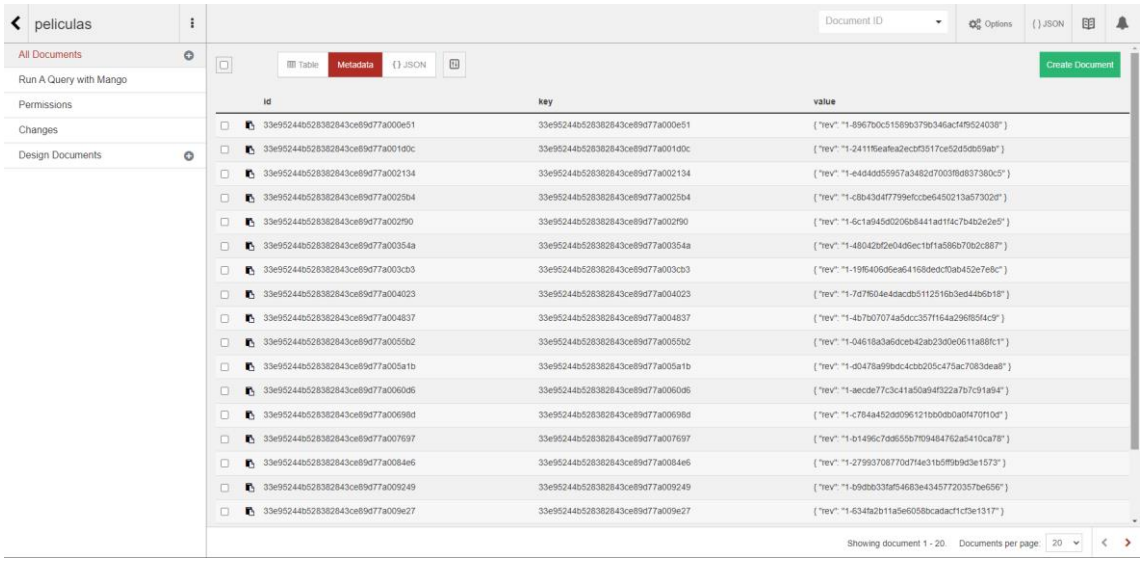
In [42]:
```
import csv
import json
def json_P(row):
    doc={}
    for key, value in row.items():
        doc[key]=value
    return doc

couch=couchdb.Server('http://admin:admin@127.0.0.1:5984/')
db=couch['peliculas']

with open("imdb_movies.csv", "r",encoding="utf-8") as file:
    reader = csv.DictReader(file)
    for row in reader:
        json_data=json_P(row)
        db.save(json_data)
```

In [40]:
```
db=couch['peliculas']
```

In [ ]:

In [ ]:



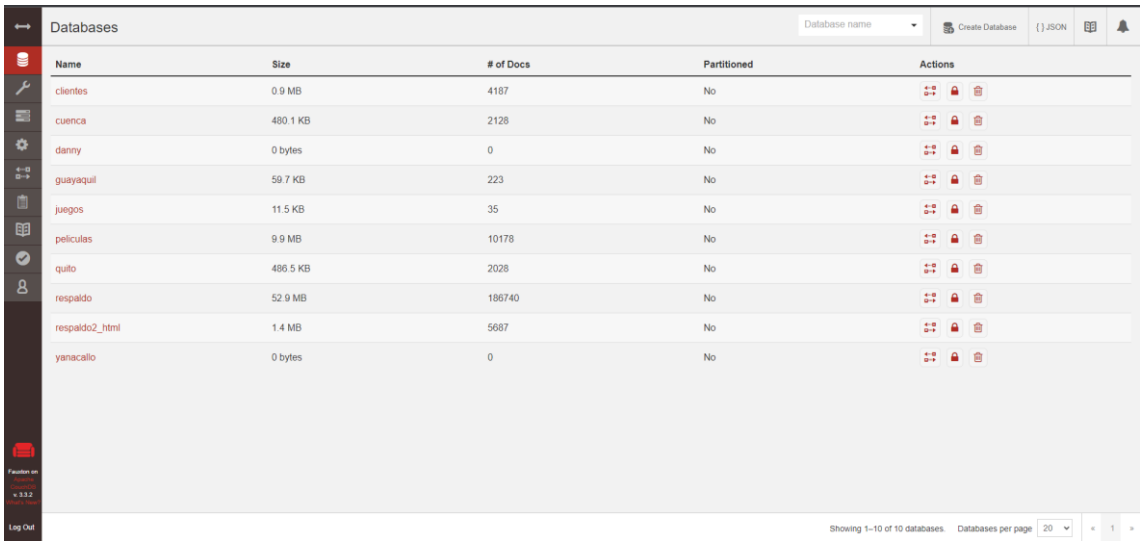


Codigo:

```
#Importar librerias

import couchdb

#conectarse a la base de datos
couch=couchdb.Server('http://admin:admin@127.0.0.1:5984/')
#crear un nuevo database
db=couch.create('peliculas')

#Vamos a trabajar con ese database
db=couch['peliculas']

#Instalar librerias
pip install csv
pip install json
#importar librerías
import csv
import json
# Transformación de cada elemento del csv en json
def json_P(row):
    doc={}
    for key, value in row.items():
        doc[key]=value
    return doc
couch=couchdb.Server('http://admin:admin@127.0.0.1:5984/')
db=couch['peliculas']

with open("imdb_movies.csv", "r",encoding="utf-8") as file:
    reader = csv.DictReader(file)
    for row in reader:
```

```
        json_data=json_P(row)
#Guardar esos elementos en la base de datos

        db.save(json_data)
```

3.- Creación de una réplica tipo “one time” en un sentido (1 punto)



Source

Type: Local database

Name: peliculas

Authentication: Username and password

admin

•••••

Target

Type: Existing local database

Name: danny

Authentication: Username and password

admin

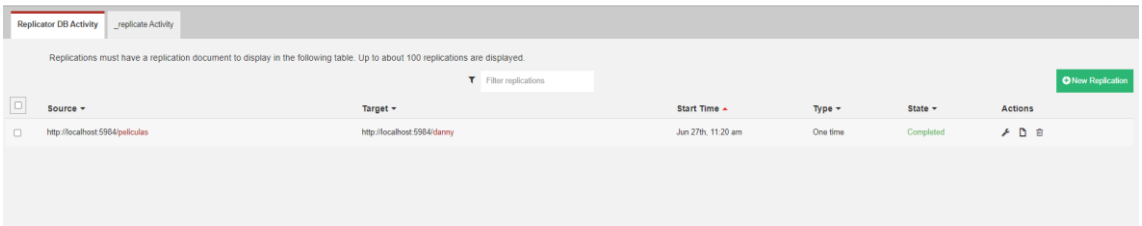
•••••

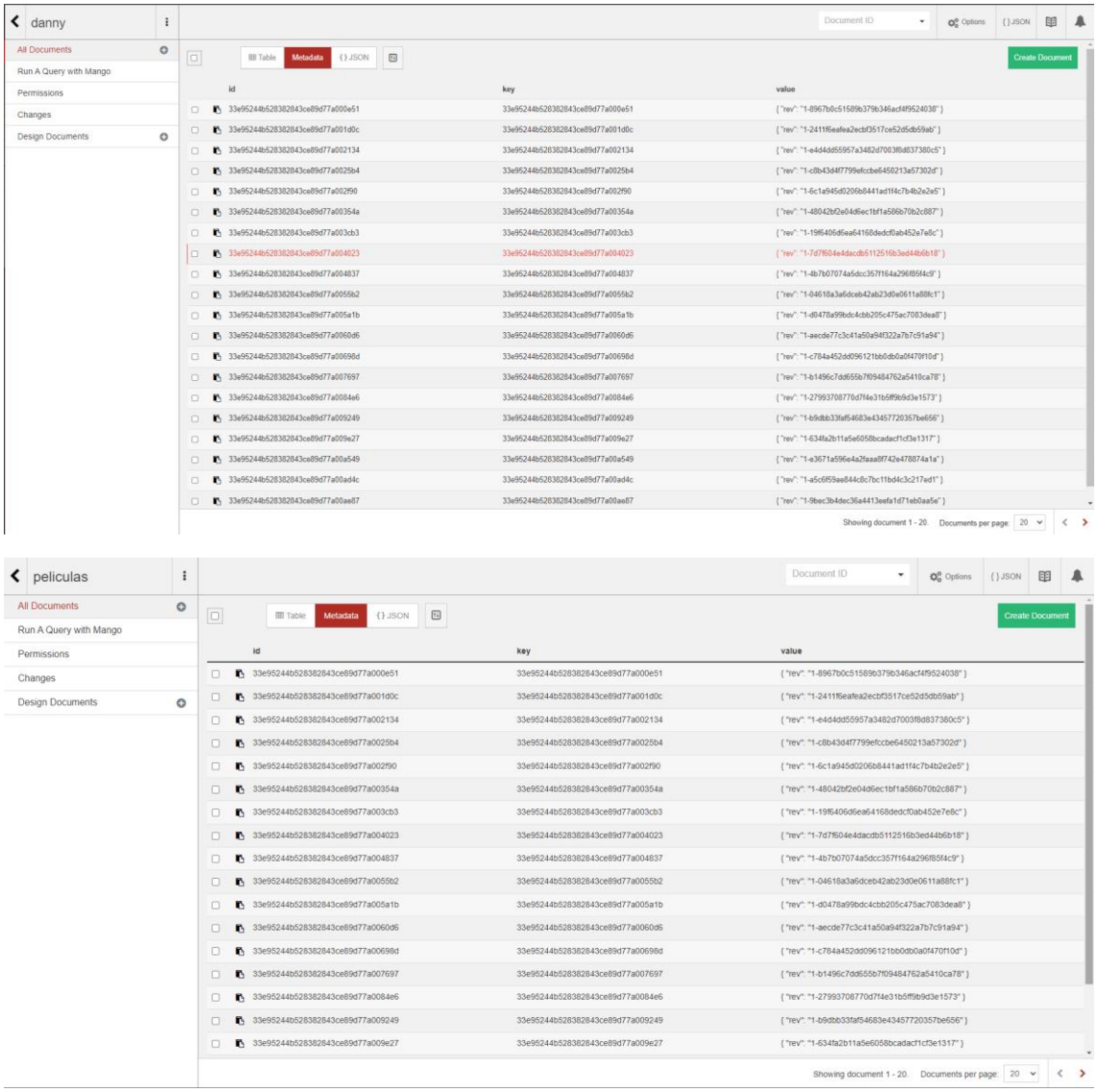
Options

Replication type: One time

Replication document: Custom ID (optional)

Start Replication Clear





4.- Creación de una réplica tipo “Continuous” en un sentido (1 punto)

Source

Type: Local database

Name: peliculas

Authentication: Username and password

admin

•••••

Target

Type: Existing local database

Name: yanacallo

Authentication: Username and password

admin

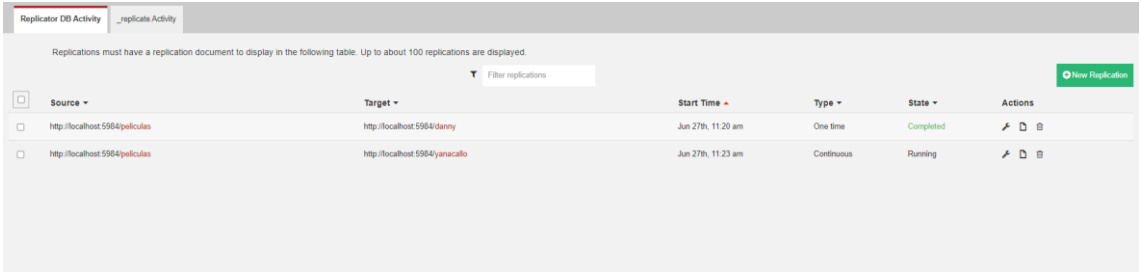
•••••

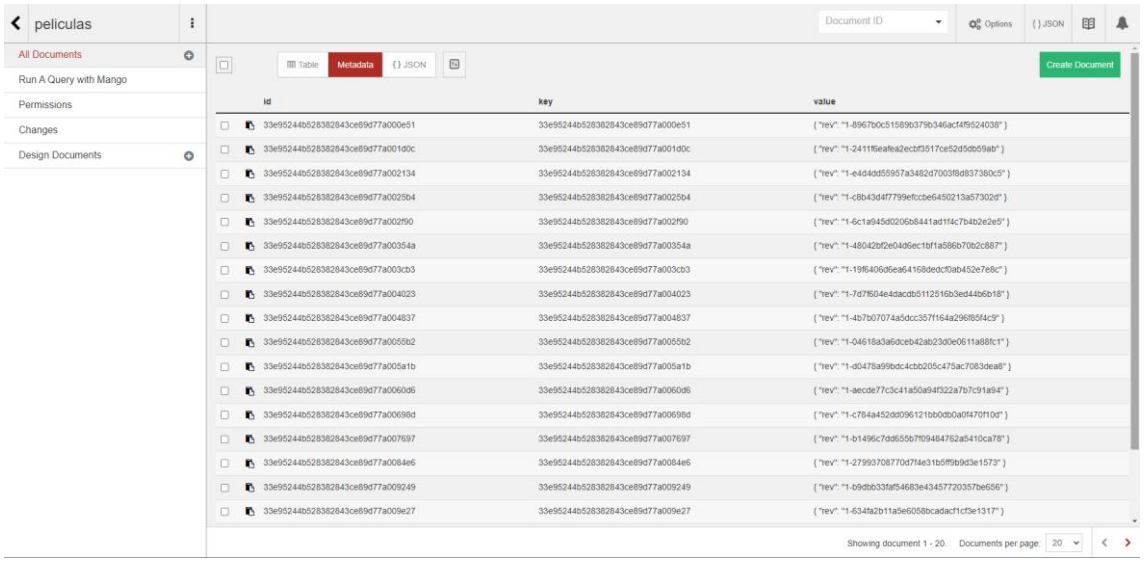
Options

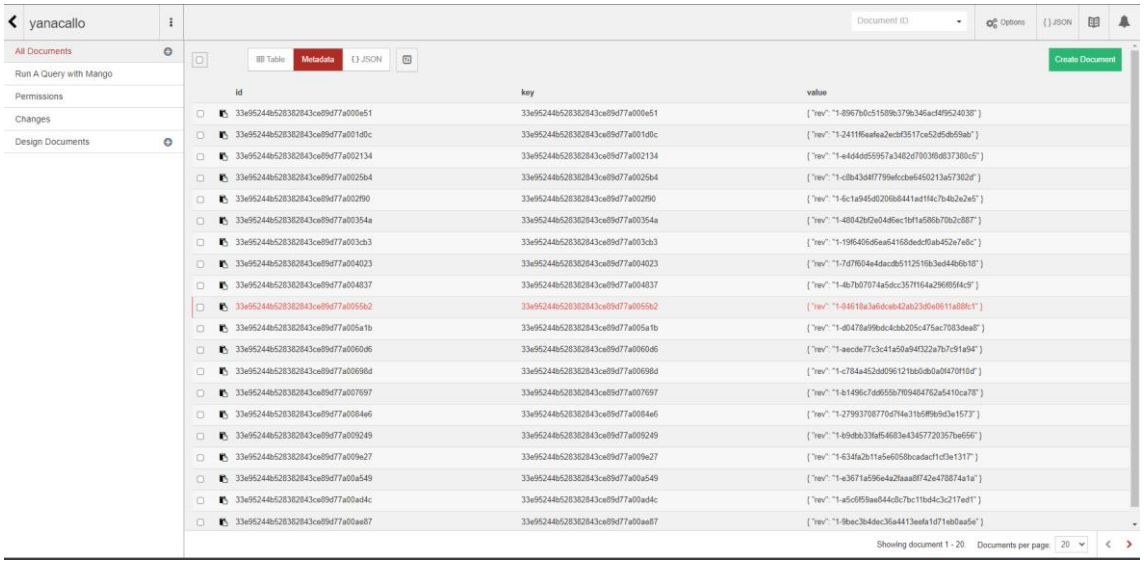
Replication type: Continuous

Replication document: Custom ID (optional)

Start Replication Clear







5.- Creación de una réplica tipo “Continuous” en dos sentidos (2 punto)

Source

Type: Local database

Name: danny

Authentication: Username and password

admin

•••••

Target

Type: Existing local database

Name: yanacallo

Authentication: Username and password

admin

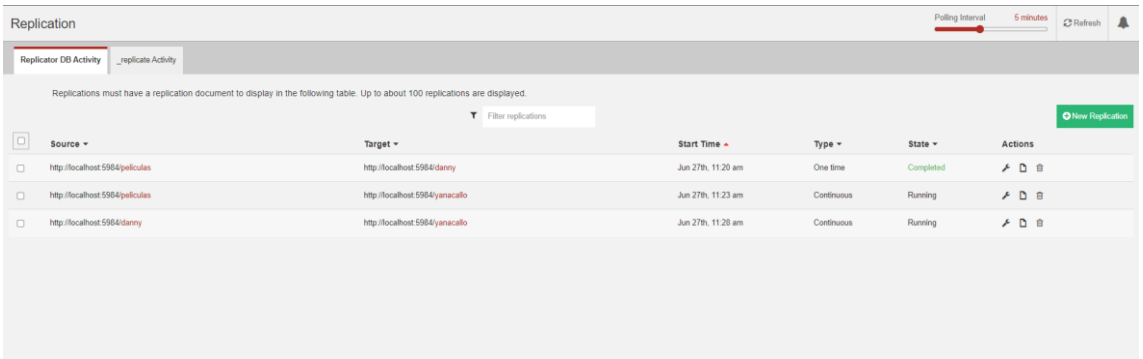
•••••

Options

Replication type: Continuous

Replication document: Custom ID (optional)

Start Replication Clear



Source

Type: Local database

Name: yanacallo

Authentication: Username and password

admin

•••••

Target

Type: Existing local database

Name: danny

Authentication: Username and password

admin

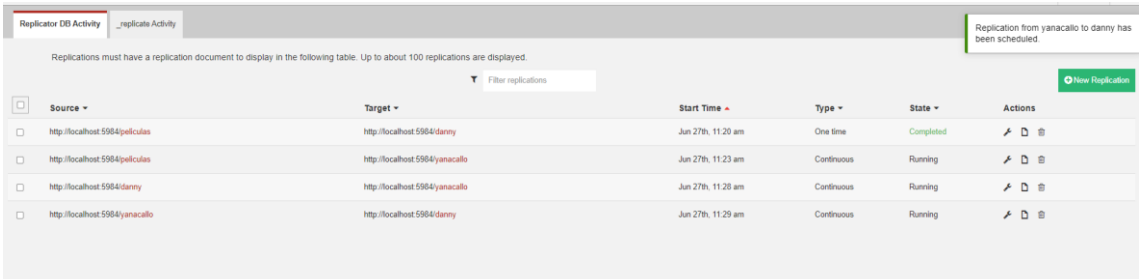
•••••

Options

Replication type: Continuous

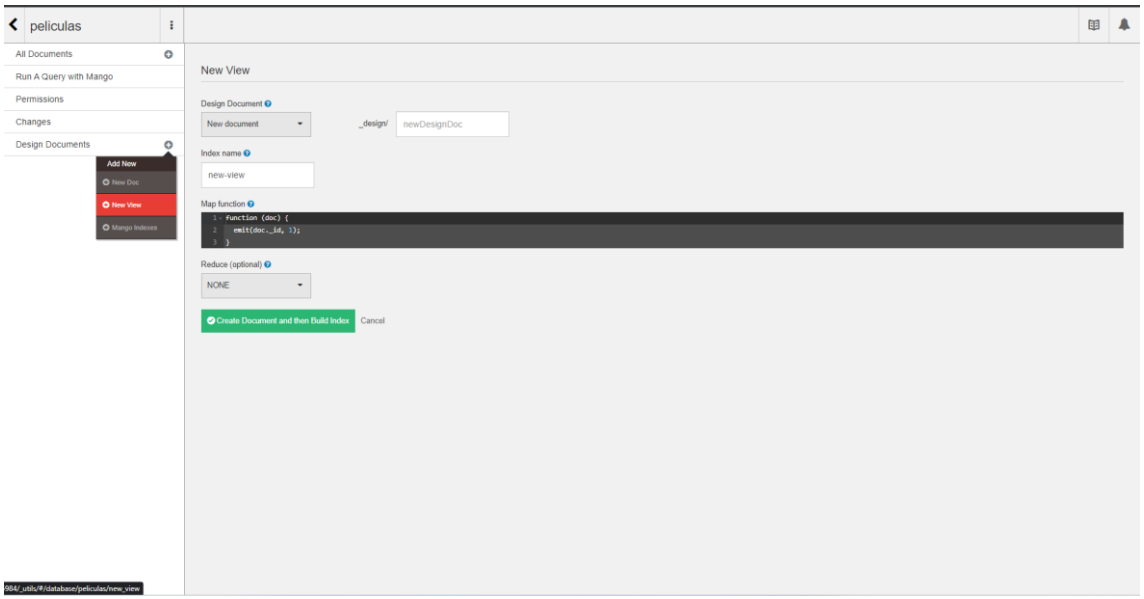
Replication document: Custom ID (optional)

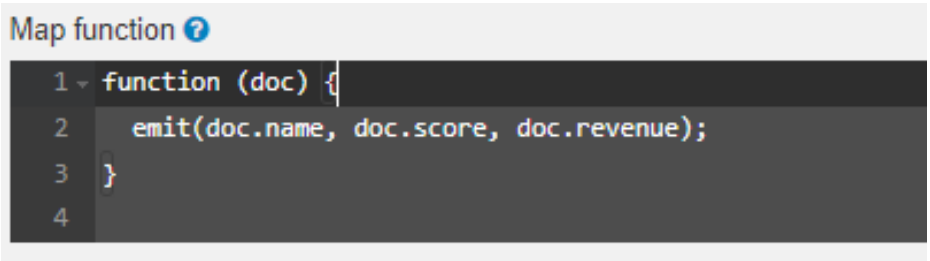
Start Replication Clear

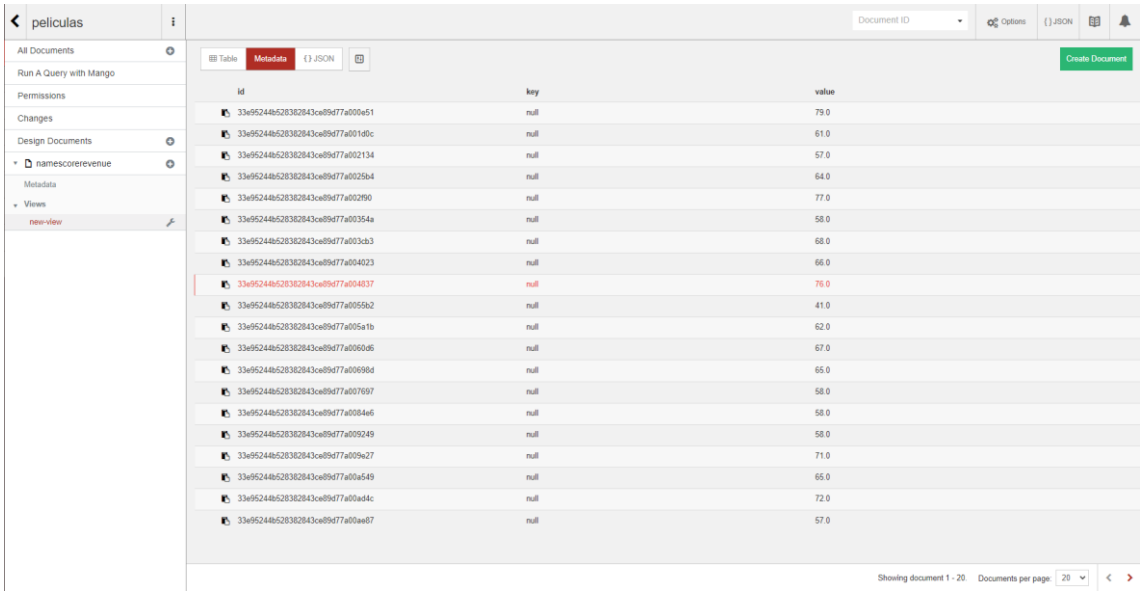


☐	http://localhost:5984/danny	http://localhost:5984/yanacallo
☐	http://localhost:5984/yanacallo	http://localhost:5984/danny

6.- Creación de una vista en CouchDB que retorne score, name y revenue (4 puntos)







7.- Exportación del json del item anterior (2 puntos)