Ejercicios Talend

MDA (EDEM). Curso 2021/2022

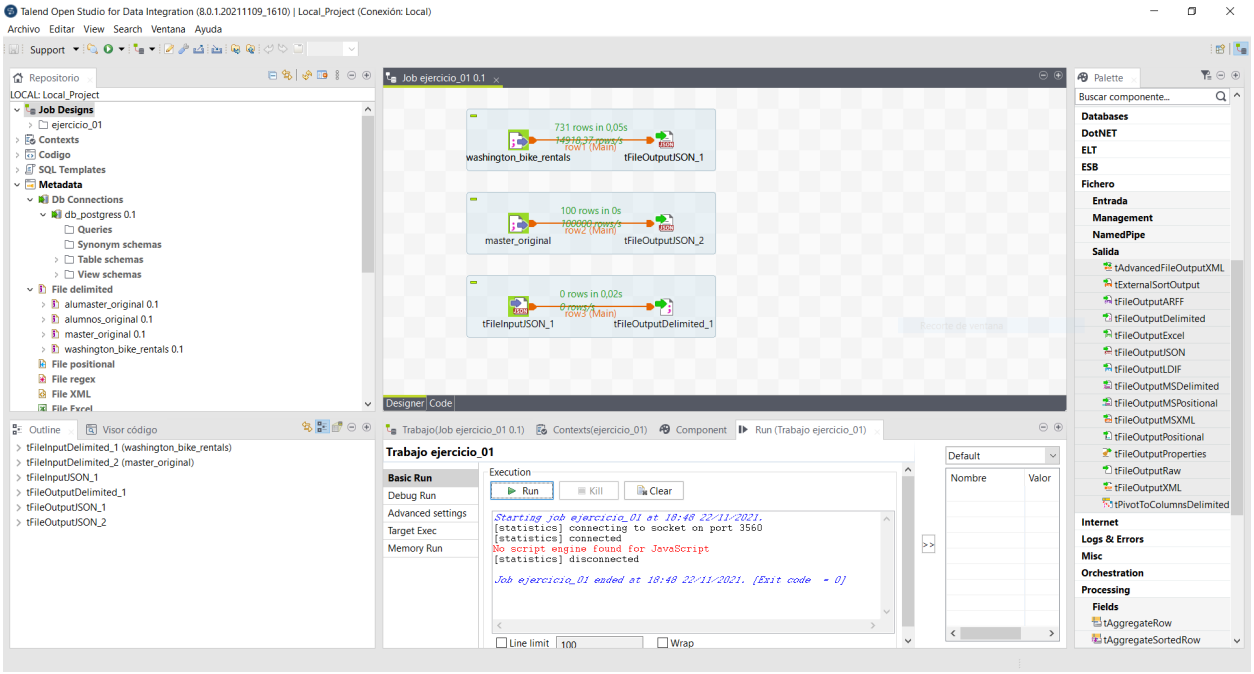
Lluna Sanz Montrull

23/11/2021

Ejercicio 1

Leer un fichero CSV y escribirlo a fichero Json en la misma carpeta. Hacer lo mismo con MASTERS.csv. Leer de un json y escribir en un CSV. - Comentario: se producen incompatibilidades con las versiones de Java al pasar un fichero de .csv a .json.

Talend



JSON

Washington bike rentals


```
{
  "data": [
    {
      "hum": "80,5833",
      "mnth": 1,
      "weekday": 6,
      "cnt": 985,
      "registered": 654,
      "holiday": 0,
      "instant": 1,
      "weathersit": 2,
      "workingday": 0,
      "dteday": "01\/01\/2011",
      "casual": 331,
      "atemp": "0,363625",
      "season": 1,
      "yr": 0,
      "temp_celsius": "14,110847",
      "windspeed_kh": "10,749882"
    },
    {
      "..."
    },
    {
      "hum": "57,75",
      "mnth": 12,
      "weekday": 1,
      "cnt": 2729,
      "registered": 2290,
      "holiday": 0,
      "instant": 731,
      "weathersit": 2,
      "workingday": 1,
      "dteday": "31\/12\/2012",
      "casual": 439,
      "atemp": "0,223487",
      "season": 1,
      "yr": 1,
      "temp_celsius": "8,849153",
      "windspeed_kh": "10,374682"
    }
  ]
}
```


731 Rows

Masters


```
{
  "data": [
    {
      "id": 1,
      "Nom": "Latz"
    },
    {
      "id": 2,
      "Nom": "Janyx"
    },
    {
      "id": 3,
      "Nom": "Eabox"
    },
    {
      "id": 4,
      "Nom": "Izio"
    },
    {
      "id": 5,
      "Nom": "Browsebug"
    },
    {
      "..."
    },
    {
      "id": 96,
      "Nom": "Realpoint"
    },
    {
      "id": 97,
      "Nom": "Skyble"
    },
    {
      "id": 98,
      "Nom": "Kimia"
    },
    {
      "id": 99,
      "Nom": "Skiba"
    },
    {
      "id": 100,
      "Nom": "Blognation"
    }
  ]
}
```


100 Rows

Ejercicio 2

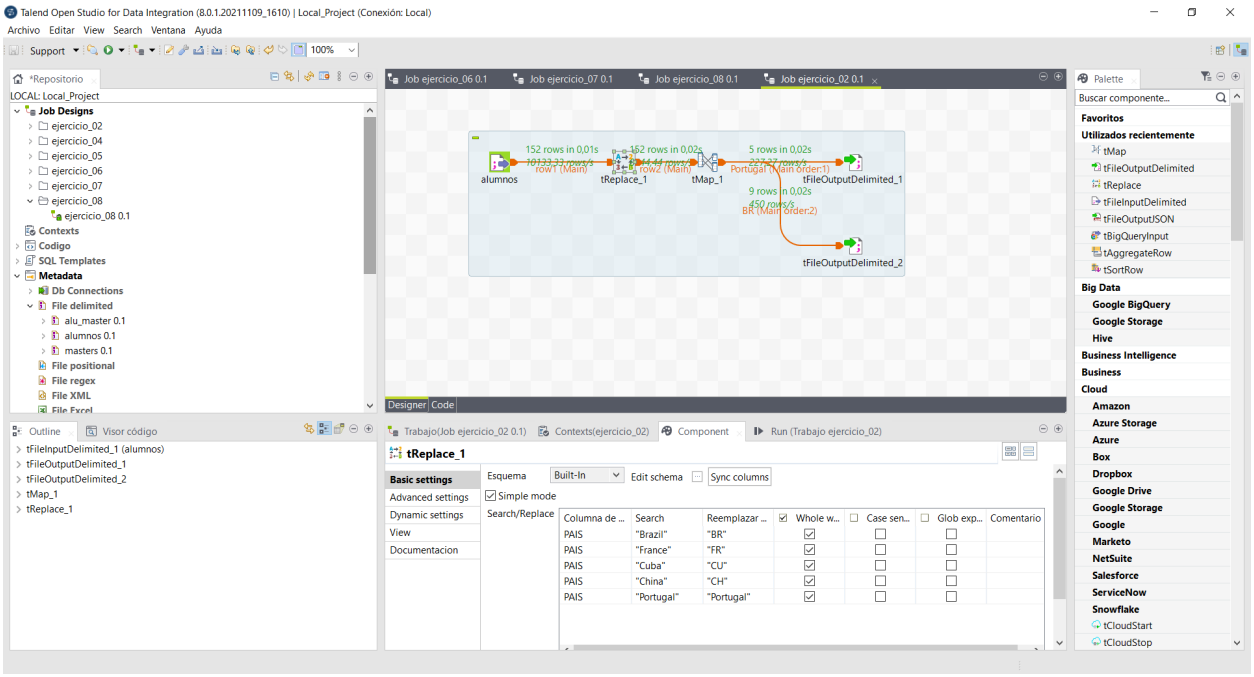
Debéis leer un fichero CSV(alumnos) y reemplazar: - China –> CH - France -> FR - Brazil -> BR - Cuba -> CU Debéis leer un fichero CSV(alumnos) y filtrar, de forma que solo me devuelva los alumnos que son de Portugal. Leer CSV ALUMNOS, reemplazar Brazil por BR, y filtrar solo los de BR.

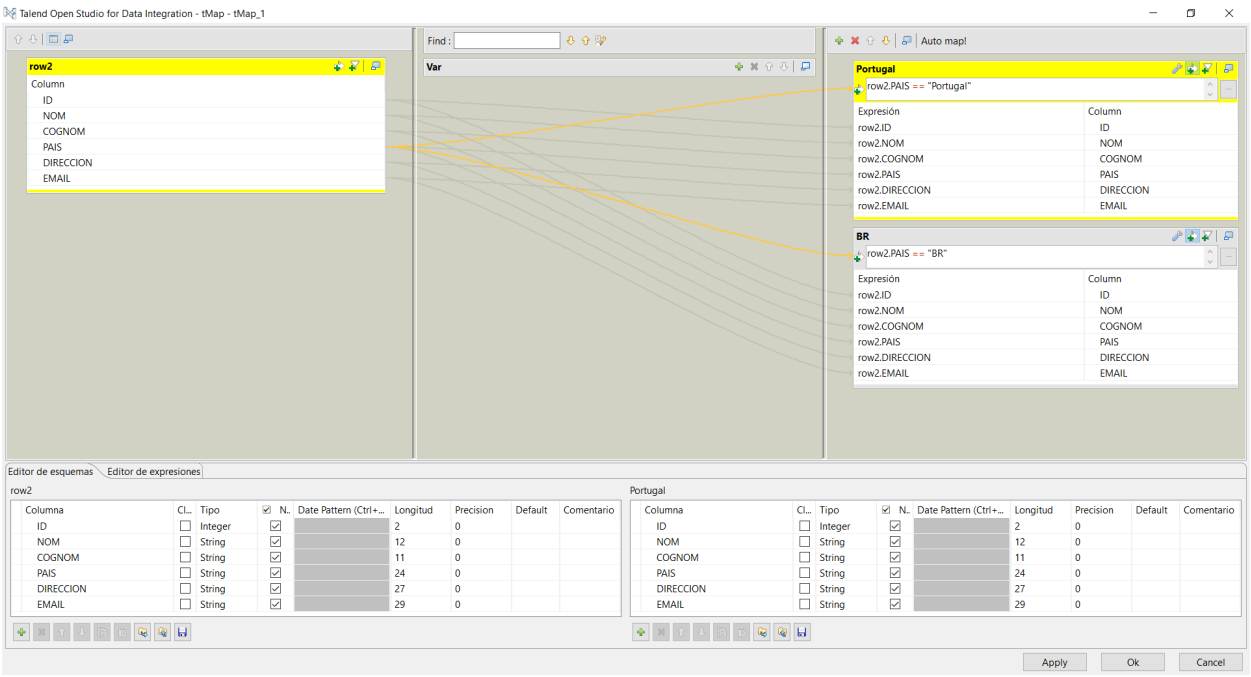
SQL

```
SELECT ID, NOM, COGNOM, REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(PAIS,
'Portugal', 'Portugal'), 'France', 'FR'), 'Cuba', 'CU'), 'China', 'CH'),
'Brazil', 'BR'), DIRECCION, EMAIL
FROM SQL_session.ALUMNOS
WHERE PAIS = "Portugal"
```

```
SELECT ID, NOM, COGNOM, REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(PAIS,
'Portugal', 'Portugal'), 'France', 'FR'), 'Cuba', 'CU'), 'China', 'CH'),
'Brazil', 'BR'), DIRECCION, EMAIL
FROM SQL_session.ALUMNOS
WHERE PAIS = "BR"
```

Talend





CSV

Alumnos de Portugal

	ID	NOM	COGNOM	PAIS	DIRECCION	EMAIL
0	1	Noëlla	Orchart	Portugal	54 Veith Parkway	rorchart0@huffingtonpost.com
1	15	Hélèna	Brooke	Portugal	1068 Sherman Crossing	tbrookee@icio.us
2	45	Marylène	Duddy	Portugal	485 Delladonna Terrace	oduddy18@samsung.com
3	96	Bénédicte	Bampforth	Portugal	5 Blaine Road	cbampforth2n@imageshack.us
4	124	Yú	Keaves	Portugal	423 Spenser Alley	akeaves3f@chicagotribune.com

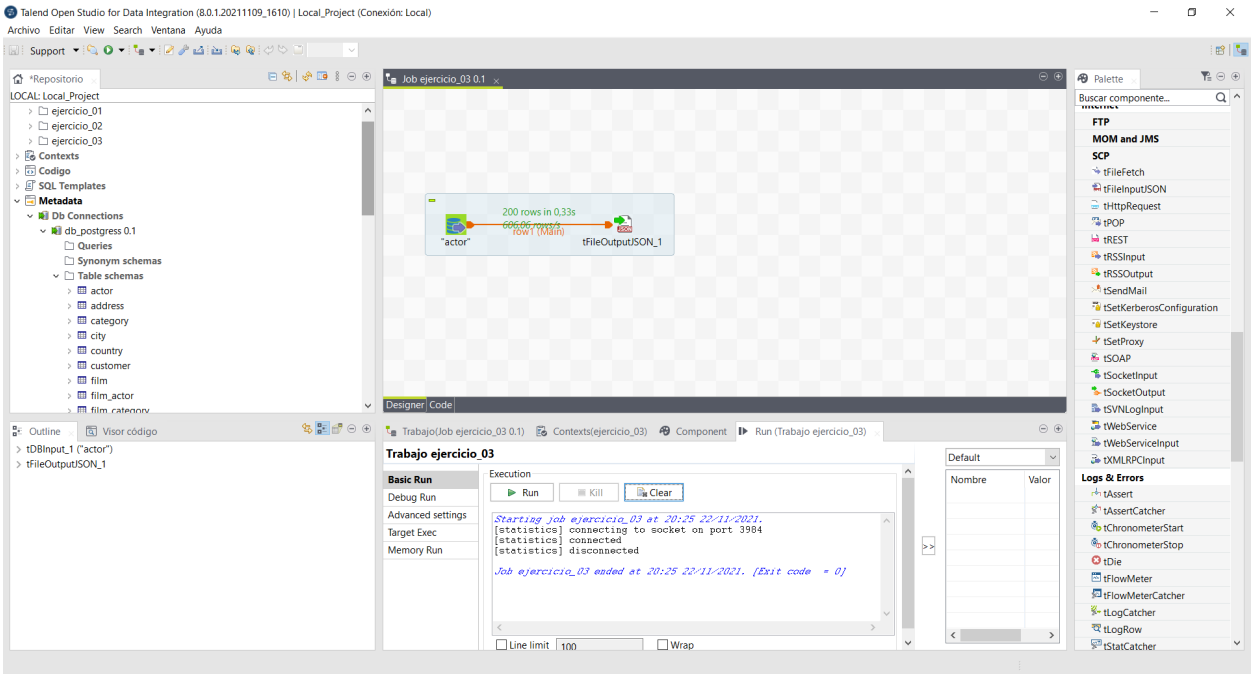
Alumnos de Brasil

	ID	NOM	COGNOM	PAIS	DIRECCION	EMAIL
0	13	Séverine	Heathcoat	BR	1602 Welch Way	jheathcoatc@wisc.edu
1	22	Maëlann	Clavering	BR	29120 David Road	bclaveringl@china.com.cn
2	44	Pélagie	Arlow	BR	0 Briar Crest Pass	harlow17@netlog.com
3	74	Méng	Greenlies	BR	2 Derek Pass	rgreenlies21@storify.com
4	94	Joséphine	Treuge	BR	6543 Jackson Parkway	btreuge2l@state.tx.us

Ejercicio 3

Leer tabla de actores y volcarlo a fichero JSON. - Instrucciones

Talend



JSON


```
{
  "data": [
    {
      "last_update": "26-05-2013",
      "last_name": "Guiness",
      "actor_id": 1,
      "first_name": "Penelope"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Wahlberg",
      "actor_id": 2,
      "first_name": "Nick"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Chase",
      "actor_id": 3,
      "first_name": "Ed"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Davis",
```

```
      "actor_id": 4,
      "first_name": "Jennifer"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Lollobrigida",
      "actor_id": 5,
      "first_name": "Johnny"
    },
    {
      "..."
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Walken",
      "actor_id": 196,
      "first_name": "Bela"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "West",
      "actor_id": 197,
      "first_name": "Reese"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Keitel",
      "actor_id": 198,
      "first_name": "Mary"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Fawcett",
      "actor_id": 199,
      "first_name": "Julia"
    },
    {
      "last_update": "26-05-2013",
      "last_name": "Temple",
      "actor_id": 200,
      "first_name": "Thora"
    }
  ]
}
```


200 rows

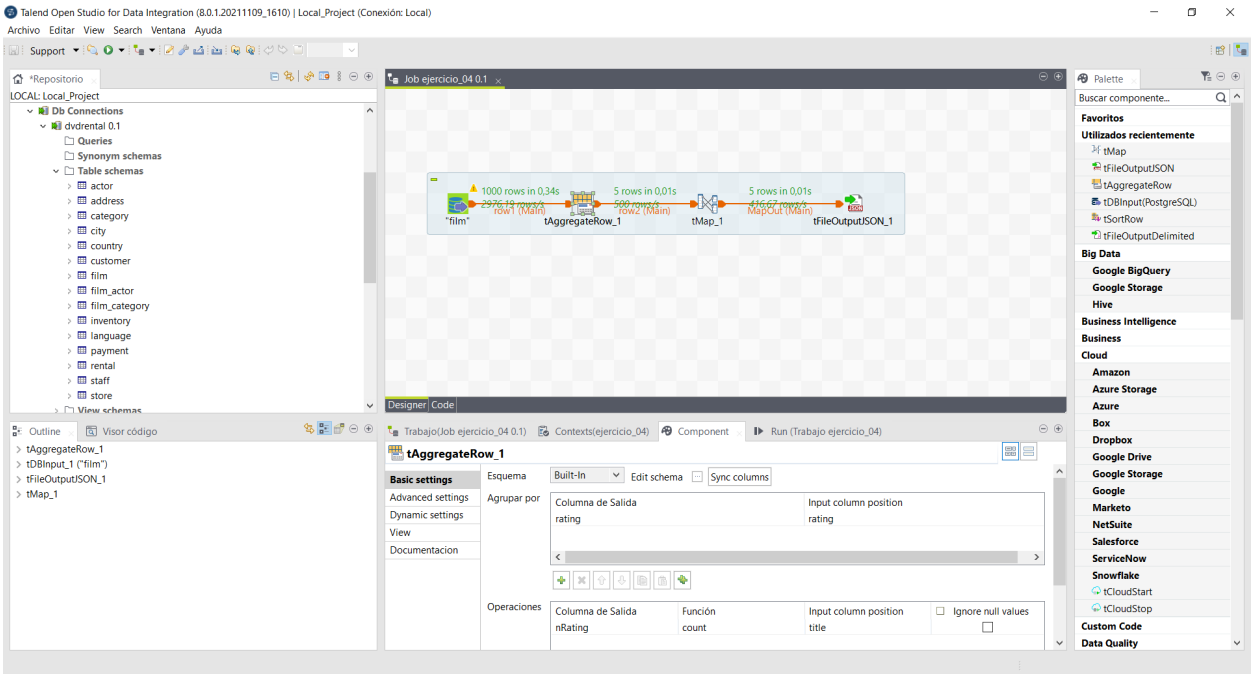
Ejercicio 4

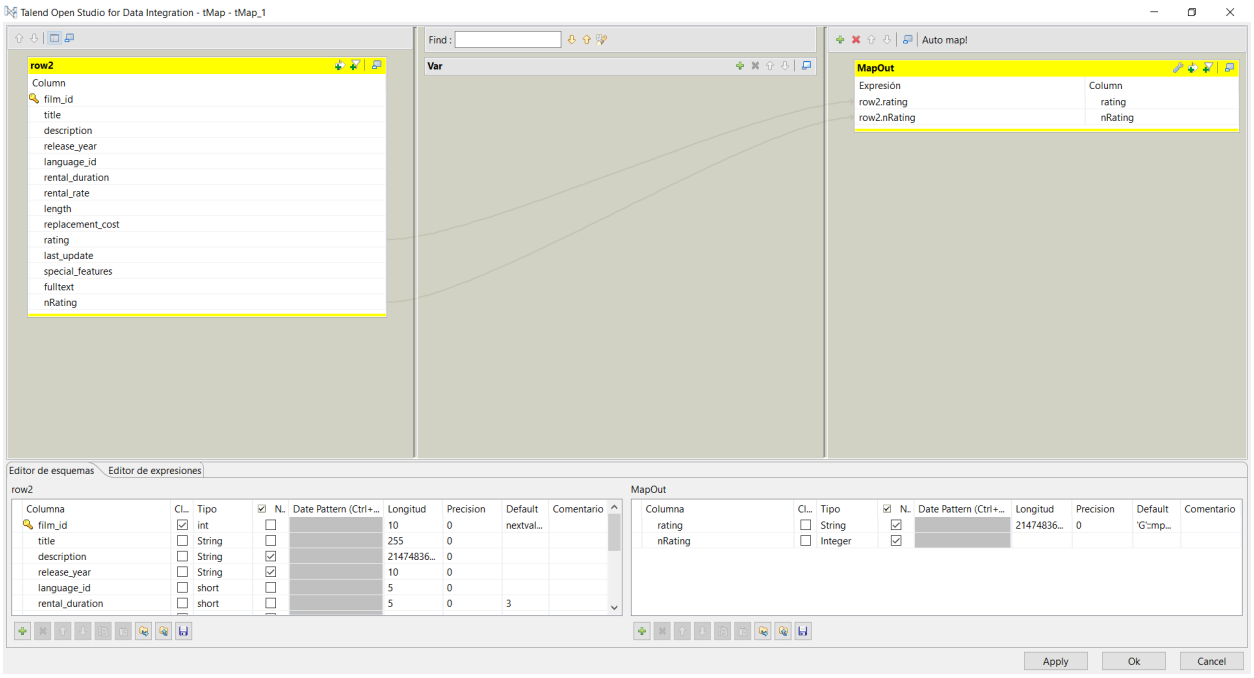
Agregar las películas por rating y mostrar un count, volcar a json el resultado

SQL

```
SELECT rating, count(title) FROM film GROUP BY rating
```

Talend





JSON


```
{
  "count_rating": [
    {
      "nRating": 195,
      "rating": "R"
    },
    {
      "nRating": 194,
      "rating": "PG"
    },
    {
      "nRating": 178,
      "rating": "G"
    },
    {
      "nRating": 210,
      "rating": "NC-17"
    },
    {
      "nRating": 223,
      "rating": "PG-13"
    }
  ]
}
```


5 Rows

Ejercicio 5

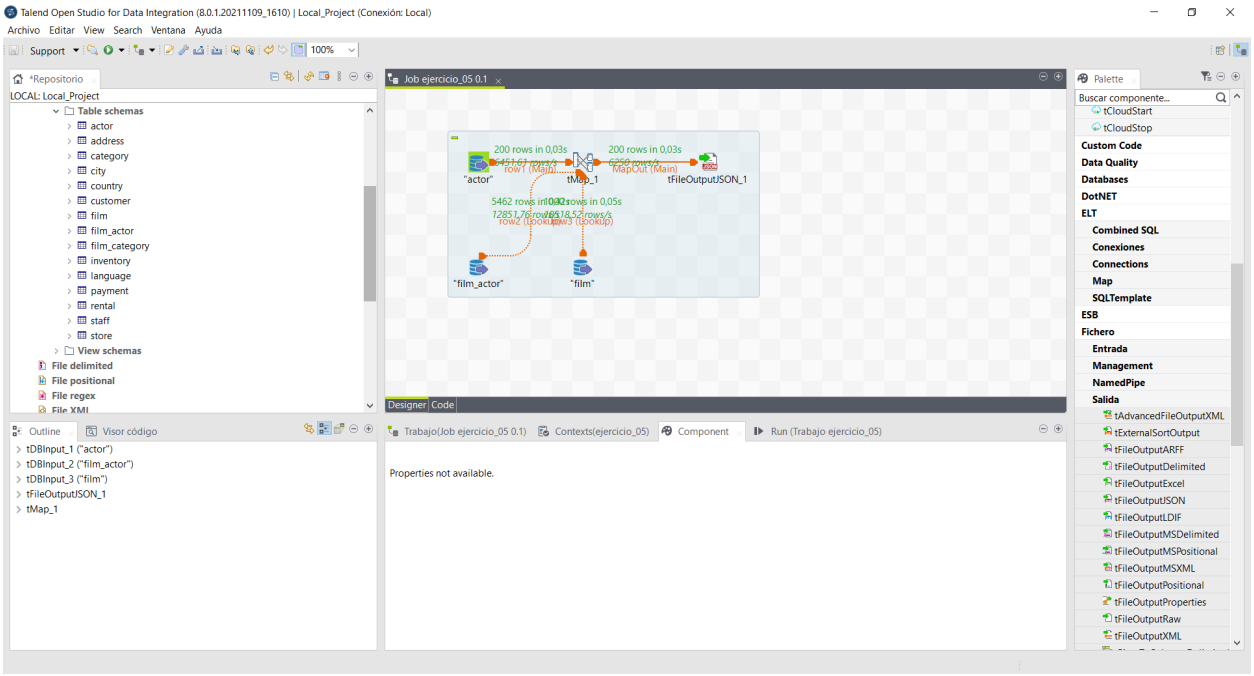
Realizar un Join entre Actor / Film / Film_Actor y volcar a json un fichero con estos campos:

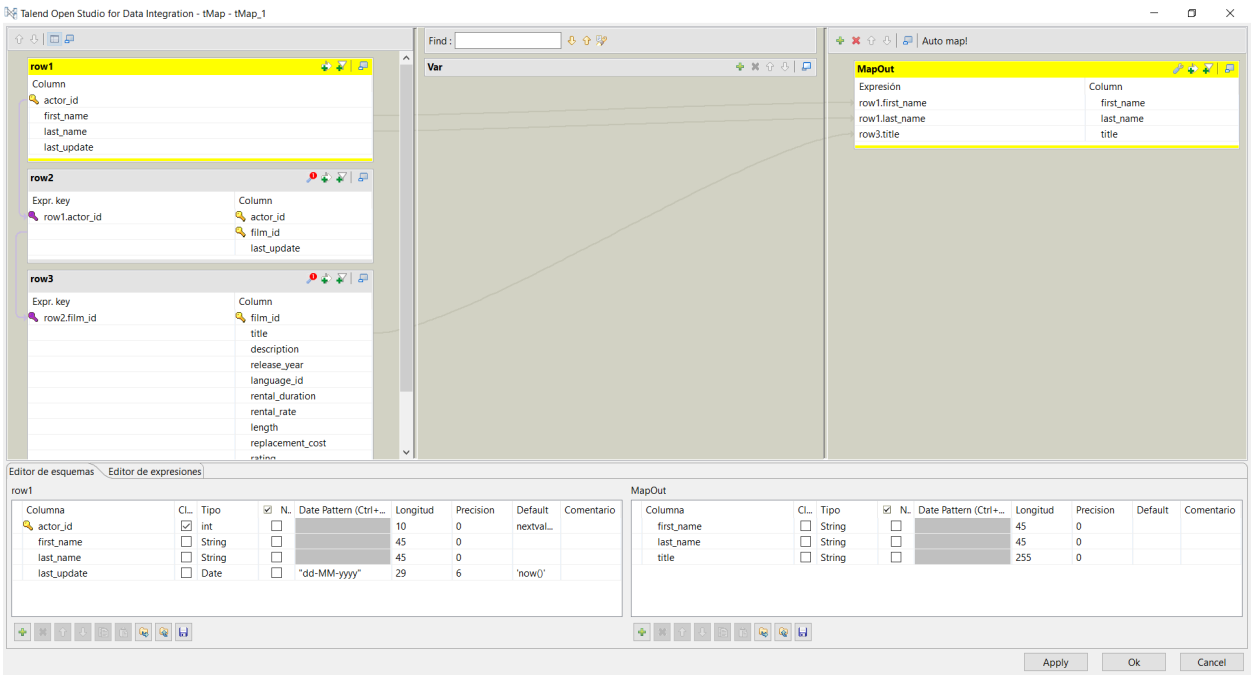
- Nombre
- Apellido
- Película

SQL

```
SELECT actor.first_name, actor.last_name, film.title
FROM actor
JOIN film_actor ON actor.actor_id=film_actor.actor_id
JOIN film ON film_actor.film_id=film.film_id
```

Talend





JSON


```
{
  "data": [
    {
      "last_name": "Guiness",
      "title": "Wizard Coldblooded",
      "first_name": "Penelope"
    },
    {
      "last_name": "Wahlberg",
      "title": "Wardrobe Phantom",
      "first_name": "Nick"
    },
    {
      "last_name": "Chase",
      "title": "Young Language",
      "first_name": "Ed"
    },
    {
      "last_name": "Davis",
      "title": "Unforgiven Zoolander",
      "first_name": "Jennifer"
    },
    {
      "last_name": "Lollobrigida",
      "title": "Sunrise League",
      "first_name": "Johnny"
    },
    {
```

```
      "..."
    },
    {
      "last_name": "Walken",
      "title": "Whisperer Giant",
      "first_name": "Bela"
    },
    {
      "last_name": "West",
      "title": "Yentl Idaho",
      "first_name": "Reese"
    },
    {
      "last_name": "Keitel",
      "title": "Youth Kick",
      "first_name": "Mary"
    },
    {
      "last_name": "Fawcett",
      "title": "Wait Cider",
      "first_name": "Julia"
    },
    {
      "last_name": "Temple",
      "title": "Wrong Behavior",
      "first_name": "Thora"
    }
  ]
}
```


200 Rows

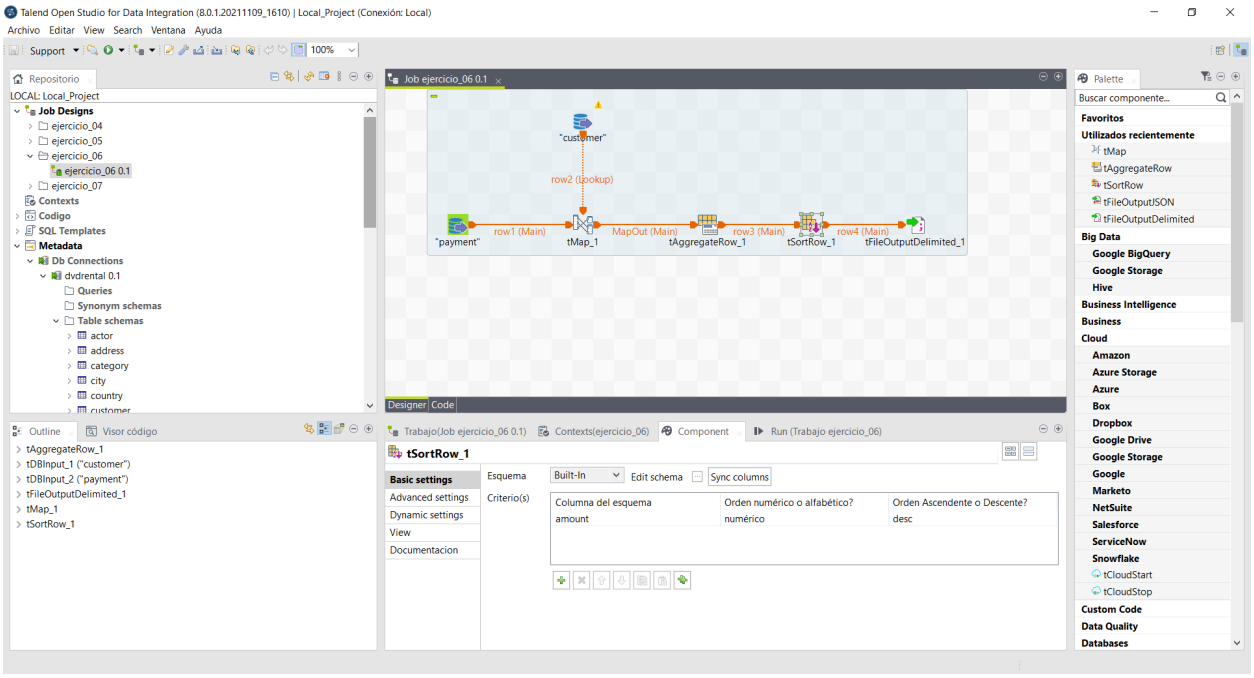
Ejercicio 6

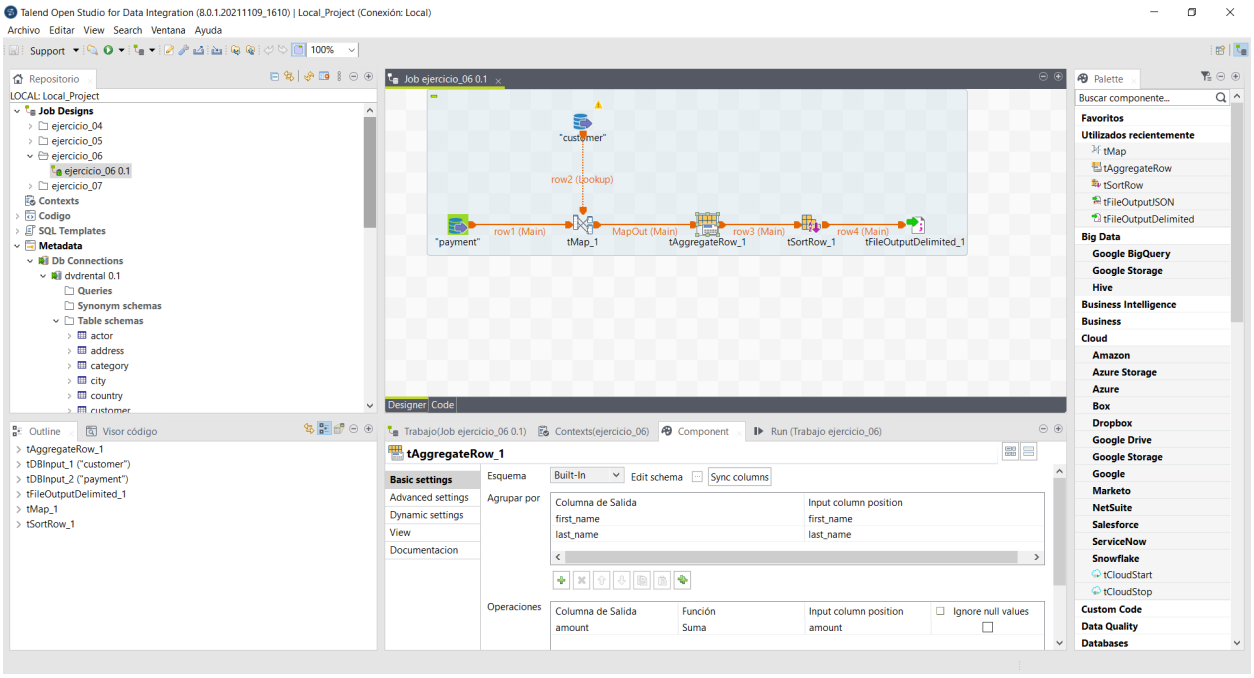
Cargar en un csv la cantidad de dinero Gastada por usuario, nombre y apellido

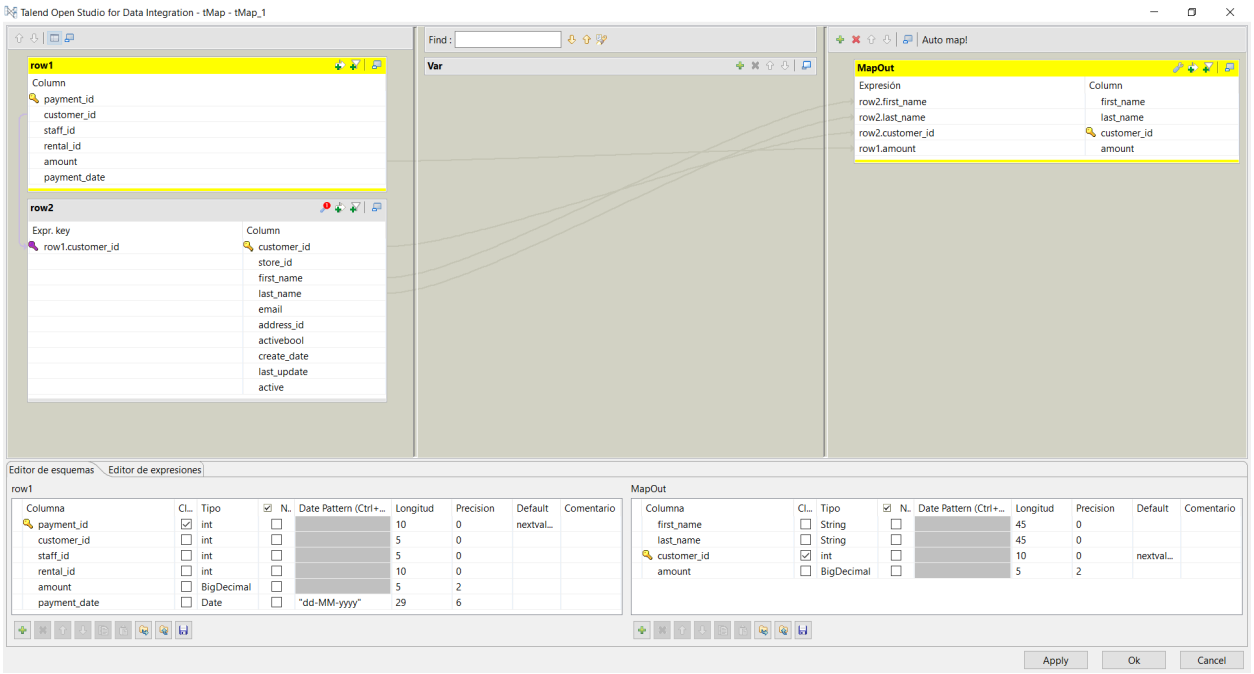
SQL

```sql
SELECT customer.first_name, customer.last_name, sum(payment.amount) FROM customer JOIN payment ON custo
```

Talend







CSV

	first_name	last_name	amount
0	Eleanor	Hunt	211.55
1	Karl	Seal	208.58
2	Marion	Snyder	194.61
3	Rhonda	Kennedy	191.62
4	Clara	Shaw	189.6
...	...	...	...
594	Tiffany	Jordan	49.88
595	Anthony	Schwab	47.85
596	Caroline	Bowman	37.87
597	Leona	Obrien	32.9
598	Brian	Wyman	27.93

599 rows × 3 columns

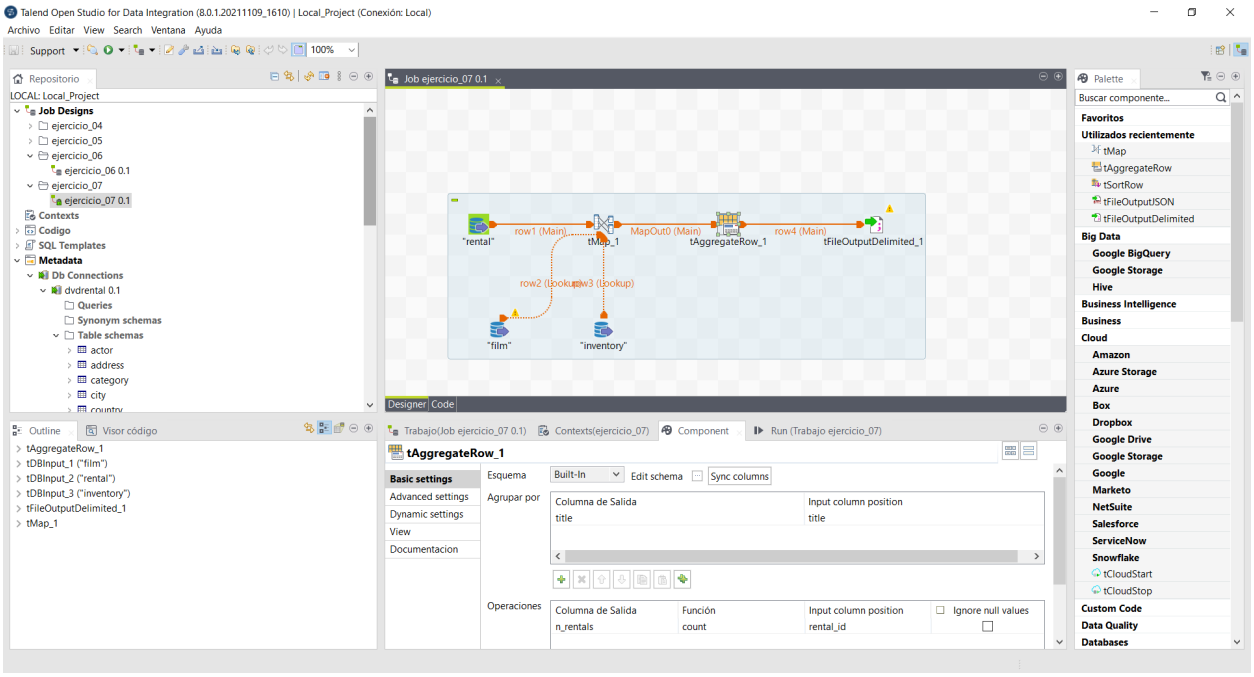
Ejercicio 7

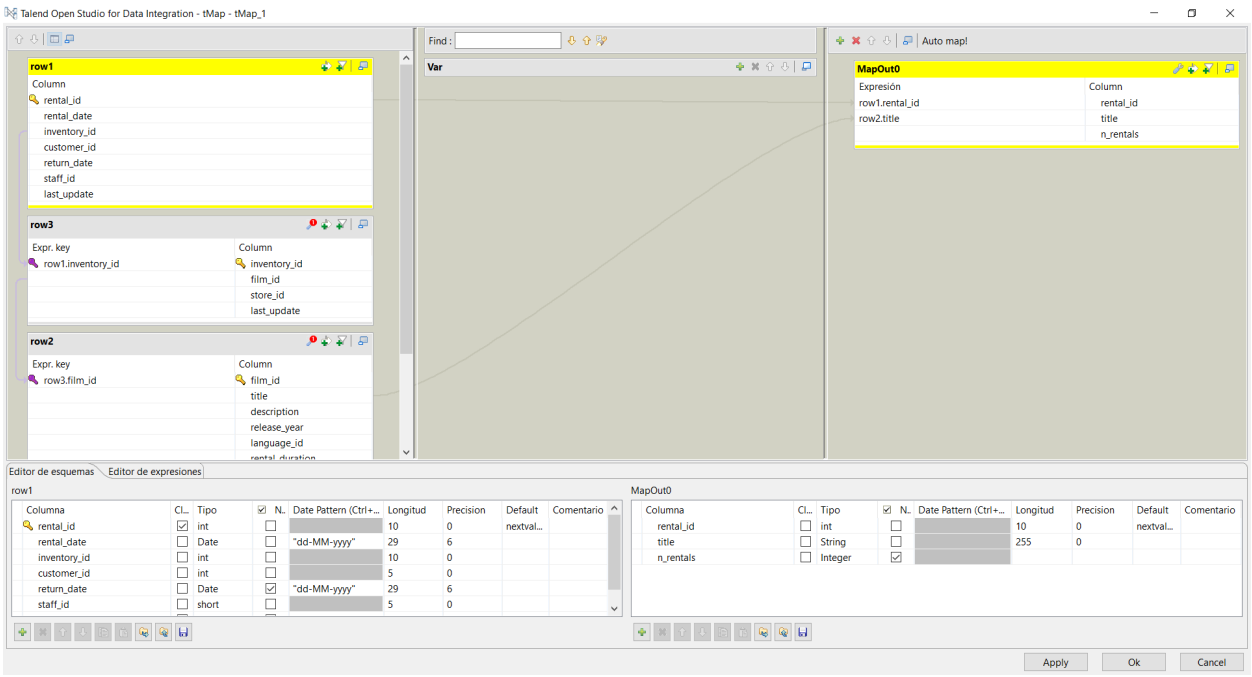
Cargar en un csv el numero de veces que se ha alquilado cada pelicula.

SQL

```
SELECT title, count(rental.rental_id)
FROM film
JOIN inventory ON film.film_id=inventory.film_id
JOIN rental ON inventory.inventory_id=rental.inventory_id
GROUP BY title
```

Talend





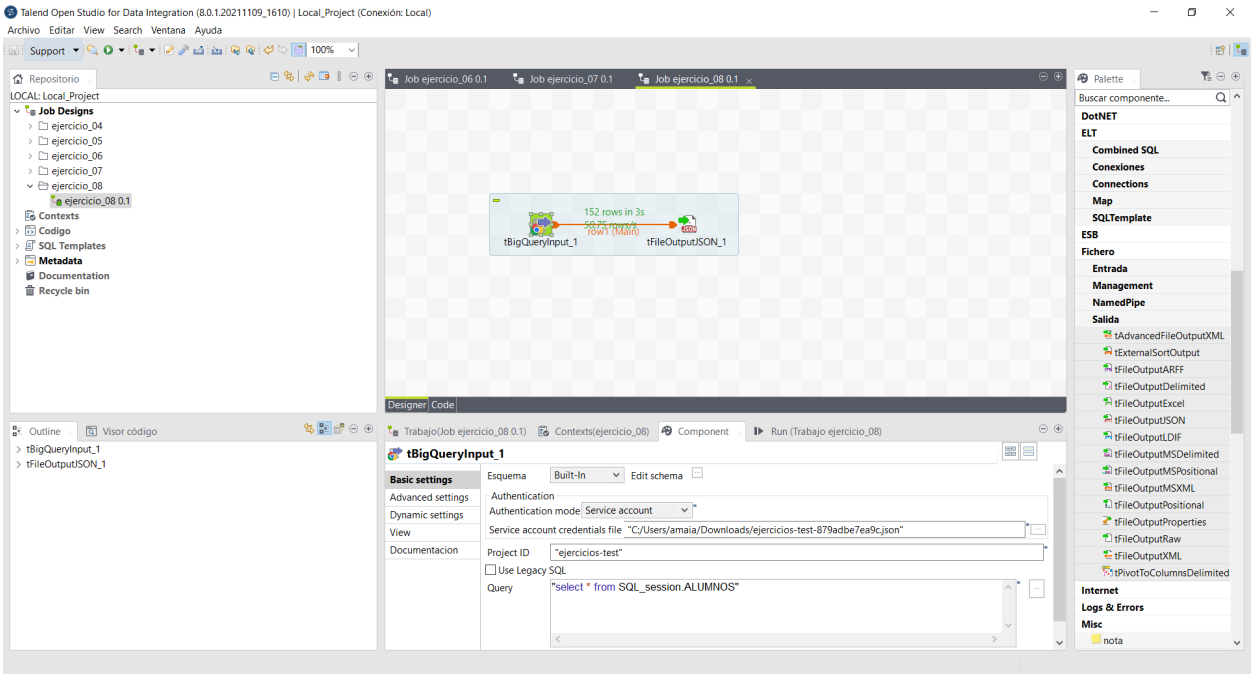
CSV

	title	rentals
0	Enough Raging	18
1	Microcosmos Paradise	13
2	Harry Idaho	30
3	Operation Operation	27
4	Gilbert Pelican	9
...	...	...
953	Vanishing Rocky	18
954	Ice Crossing	24
955	Bonnie Holocaust	16
956	Heaven Freedom	18
957	Saturday Lambs	28

958 rows × 2 columns

Ejercicio 8

Import from BigQuery



JSON


```
{
  "data": [
    {
      "DIRECCION": "16340 La Follette Place",
      "ID": 9,
      "EMAIL": "epolleye8@businessinsider.com",
      "NOM": "Lén",
      "COGNOM": "Polleye",
      "PAIS": "Cuba"
    },
    {
      "DIRECCION": "90940 Roxbury Road",
      "ID": 121,
      "EMAIL": "drugge3c@ox.ac.uk",
      "NOM": "Märta",
      "COGNOM": "Rugge",
      "PAIS": "Cuba"
    },
    {
      "DIRECCION": "3224 Mcguire Hill",
      "ID": 55,
      "EMAIL": "alukasen1i@ox.ac.uk",
      "NOM": "Marie-hélène",
      "COGNOM": "Lukasen",
      "PAIS": "Peru"
```

```
  },
  {
    "DIRECCION": "983 Cascade Trail",
    "ID": 89,
    "EMAIL": "cmoffet2g@biglobe.ne.jp",
    "NOM": "Amélie",
    "COGNOM": "Moffet",
    "PAIS": "Peru"
  },
  {
    "DIRECCION": "5 New Castle Alley",
    "ID": 3,
    "EMAIL": "mcatherick2@yale.edu",
    "NOM": "Eliès",
    "COGNOM": "Catherick",
    "PAIS": "China"
  },
  {
    "..."
  },
  {
    "DIRECCION": "7 Manitowish Hill",
    "ID": 148,
    "EMAIL": "ttaggart43@hhs.gov",
    "NOM": "Dà",
    "COGNOM": "Taggart",
    "PAIS": "Czech Republic"
  },
  {
    "DIRECCION": "73 Green Ridge Hill",
    "ID": 114,
    "EMAIL": "vgoudie35@constantcontact.com",
    "NOM": "Miléna",
    "COGNOM": "Goudie",
    "PAIS": "United Kingdom"
  },
  {
    "DIRECCION": "0 Miller Road",
    "ID": 95,
    "EMAIL": "gcommucci2m@buzzfeed.com",
    "NOM": "Thérèse",
    "COGNOM": "Commucci",
    "PAIS": "Papua New Guinea"
  },
  {
    "DIRECCION": "56 Londonderry Road",
    "ID": 135,
    "EMAIL": "astert3q@aboutads.info",
    "NOM": "Cécilia",
    "COGNOM": "Stert",
    "PAIS": "Bosnia and Herzegovina"
  },
  {
```

```
      "DIRECCION": "5 Arizona Crossing",
      "ID": 32,
      "EMAIL": "eglisenanv@technorati.com",
      "NOM": "Véronique",
      "COGNOM": "Glisenan",
      "PAIS": "Central African Republic"
    }
  ]
}
```


152 Rows