

DEPLOYMENT

Database

Data LakePREVIEW

DATA SERVICES

Triggers

Data API

Data Federation

SECURITY

Database Access

Network Access

Advanced

Atlas Milestones

✓

Complete

DATABASES: 1COLLECTIONS: 1

VISUALIZE YOUR DATA

DATA MODEL EXAMPLES

REFRESH

+ Create Database

Search Namespaces

Tienda

Ventas

Tienda.Ventas

STORAGE SIZE: 36KBTOTAL DOCUMENTS: 5INDEXES TOTAL SIZE: 36KB

FindIndexesSchema Anti-Patterns 0AggregationSearch Indexes ●

INSERT DOCUMENT

FILTER { field: 'value' }

OPTIONSApplyReset

QUERY RESULTS: 1-4 OF 4

_id: ObjectId("62d1378d1b23c179142ab016")

nombreCliente: "Victor Calvache"

Catan: 35

total: 35

_id: ObjectId("62d13ccd1b23c179142ab018")

nombreCliente: "Patti Smith"

Arkham Horror: 65

total: 65

_id: ObjectId("62d13d981b23c179142ab019")

nombreCliente: "PJ Harvey"

Exploding kittens: 19

total: 19

_id: ObjectId("62d13de81b23c179142ab01a")

nombreCliente: "Victor Calvache"

Zombicide: 99

total: 124

7Wondes Duel: 25

Descripció:

Comencem a familiaritzar-nos amb bases de dades NoSQL !!! Comencem amb uns quants exercicis bàsics

```
In [1]: import pymongo
        from pymongo.server_api import ServerApi
        from datetime import datetime
        from pprint import pprint
        from IPython.display import Image
        from pymongo import MongoClient
        import pandas as pd
        import matplotlib.pyplot as plt
        import numpy as np
        pd.set_option('display.max_columns', None)
```

Nivell 1

- Exercici 1:

Crea una base de dades NoSQL utilitzant MongoDB. Afegeix-li algunes dades d'exemple que et permetin comprovar que ets capaç de processar-ne la informació de manera bàsica.

```
In [2]: # Base de datos creada con Atlas, el servicio nube de MongoDB
        Image(url="mongo_db.jpeg", width=7000, height=7000)
```



- Exercici 2:

Connecta la base de dades NoSQL a Python utilitzant per exemple pymongo.

```
In [3]: # Conexión a la base de datos
        myclient = pymongo.MongoClient("mongodb+srv://admin:admin1234@cluster0.nd3jk3l.mongodb.net/?retryWrites=true&w=
```

```
In [4]: # Print de comprobación en el que mostramos las bases de datos disponibles en el cliente conectado.
        print(myclient.list_database_names())
```

```
['Tienda', 'admin', 'local']
```

```
In [5]: # Selecciono la base de datos deseada.
        mydb = myclient["Tienda"]
```

```
In [6]: print(mydb.list_collection_names())
```

```
['Ventas']
```

```
In [7]: # Selección de la colección deseada y print para ver su primer elemento.
        ticket = mydb["Ventas"]
        x = ticket.find_one()
        print("Listado de tickets:\n",x)
```

```
Listado de tickets:
{'_id': ObjectId('62d1378d1b23c179142ab016'), 'nombreCliente': 'Victor Calvache', 'Catan': 35, 'total': 35}
```

```
In [8]: # Genero una nueva venta, que se añadirá a la colección. Incluye los datos a rellenar
        # para el recibo del cliente.
        new_sale = {
            'nombreCliente': 'Victor Calvache',
            'Orbital': 25,
            'total': 25
        }
```

```
In [9]: new_sale2 = {
            'nombreCliente': 'Iggy Pop',
            'Orbital': 25,
            'total': 25
        }
```

```
In [10]: # Inserto la nueva entrada en la colección. Se imprime el número de ID que tiene la nueva entrada.
        x = ticket.insert_one(new_sale)
        print(x.inserted_id)
```

```
62d16d7200e5b3c9aa5d405f
```

```
In [11]: y = ticket.insert_one(new_sale2)
        print(y.inserted_id)
```

```
62d16d7200e5b3c9aa5d4060
```

```
In [12]: # Imprimo en pantalla todos los elementos de la coleccion Tickets de ventas.
        cursor = ticket.find({})
        for document in cursor:
            pprint(document)
```

```
{'Catan': 35,
 '_id': ObjectId('62d1378d1b23c179142ab016'),
 'nombreCliente': 'Victor Calvache',
 'total': 35}
{'Arkham Horror': 65,
 '_id': ObjectId('62d13ccd1b23c179142ab018'),
 'nombreCliente': 'Patti Smith',
 'total': 65}
{'Exploding kittens': 19,
 '_id': ObjectId('62d13d981b23c179142ab019'),
 'nombreCliente': 'PJ Harvey',
 'total': 19}
{'7Wondes Duel': 25,
 'Zombicide': 99,
 '_id': ObjectId('62d13de81b23c179142ab01a'),
 'nombreCliente': 'Victor Calvache',
 'total': 124}
{'Orbital': 25,
 '_id': ObjectId('62d16d7200e5b3c9aa5d405f'),
 'nombreCliente': 'Victor Calvache',
 'total': 25}
{'Orbital': 25,
 '_id': ObjectId('62d16d7200e5b3c9aa5d4060'),
 'nombreCliente': 'Iggy Pop',
 'total': 25}
```

Nivell 2

- Exercici 1:

Carega algunes consultes senzilles a un Pandas Dataframe.

```
In [13]: # Creo una búsqueda para ver los clientes que han hecho una compra total de más de 100 euros.
        mydb = myclient["Tienda"]
        mycol = mydb["Ventas"]

        myquery = { "total": {"$gt": 100 }}

        mydoc = mycol.find(myquery)
        for x in mydoc:
            print(x)
```

```
{'_id': ObjectId('62d13de81b23c179142ab01a'), 'nombreCliente': 'Victor Calvache', 'Zombicide': 99, 'total': 124, '7Wondes Duel': 25}
```

```
In [14]: # Busco todas las ventas y lo añado a un Dataframe.
        raw_df = pd.DataFrame(list(ticket.find({})))
```

```
In [15]: raw_df
```

	_id	nombreCliente	Catan	total	Arkham Horror	Exploding kittens	Zombicide	7Wondes Duel	Orbital	
0	62d1378d1b23c179142ab016	Victor Calvache	35.0	35	NaN	NaN	NaN	NaN	NaN	NaN
1	62d13ccd1b23c179142ab018	Patti Smith	NaN	65	65.0	NaN	NaN	NaN	NaN	NaN
2	62d13d981b23c179142ab019	PJ Harvey	NaN	19	NaN	19.0	NaN	NaN	NaN	NaN
3	62d13de81b23c179142ab01a	Victor Calvache	NaN	124	NaN	NaN	NaN	99.0	25.0	NaN
4	62d16d7200e5b3c9aa5d405f	Victor Calvache	NaN	25	NaN	NaN	NaN	NaN	NaN	25.0
5	62d16d7200e5b3c9aa5d4060	Iggy Pop	NaN	25	NaN	NaN	NaN	NaN	NaN	25.0

Nivell 3

- Exercici 1:

Genera un resum estadístic de la informació que conté la base de dades.

```
In [19]: # Transformo las variables categóricas mediante un Dummy DF
        df = pd.get_dummies(raw_df, columns=["nombreCliente"])
        df
```

	_id	Catan	total	Arkham Horror	Exploding kittens	Zombicide	7Wondes Duel	Orbital	nombreCliente_Iggy Pop	nombreCliente_P. Harvey
0	62d1378d1b23c179142ab016	35.0	35	NaN	NaN	NaN	NaN	NaN	0	0
1	62d13ccd1b23c179142ab018	NaN	65	65.0	NaN	NaN	NaN	NaN	0	0
2	62d13d981b23c179142ab019	NaN	19	NaN	19.0	NaN	NaN	NaN	0	0
3	62d13de81b23c179142ab01a	NaN	124	NaN	NaN	99.0	25.0	NaN	0	0
4	62d16d7200e5b3c9aa5d405f	NaN	25	NaN	NaN	NaN	NaN	25.0	0	0
5	62d16d7200e5b3c9aa5d4060	NaN	25	NaN	NaN	NaN	NaN	25.0	1	0

```
In [17]: # Descripción de el contenido de la tabla mediante la función describe
        df.describe(include="all")
```

	_id	Catan	total	Arkham Horror	Exploding kittens	Zombicide	7Wondes Duel	Orbital	nombreCliente_Iggy Pop	nombreCliente_P. Harvey
count		6	1.0	6.000000	1.0	1.0	1.0	1.0	2.0	6.000000
unique		6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	62d1378d1b23c179142ab016	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean		NaN	35.0	48.833333	65.0	19.0	99.0	25.0	25.0	0.166667
std		NaN	NaN	40.320797	NaN	NaN	NaN	NaN	0.0	0.408248
min		NaN	35.0	19.000000	65.0	19.0	99.0	25.0	25.0	0.000000
25%		NaN	35.0	25.000000	65.0	19.0	99.0	25.0	25.0	0.000000
50%		NaN	35.0	30.000000	65.0	19.0	99.0	25.0	25.0	0.000000
75%		NaN	35.0	57.500000	65.0	19.0	99.0	25.0	25.0	0.000000
max		NaN	35.0	124.000000	65.0	19.0	99.0	25.0	25.0	1.000000

```
In [20]: # Cantidad total de las ventas
        total_sales = np.sum(raw_df['total'])
        print("Total sales: ", total_sales)
```

Total sales: 293