MongoDB vs PostgreSQL

A comprehensive analysis

Samiksha Burkul, Rachel Culbreath, and Krishnasurya Gopalakrishnan

What is MongoDB?

- It is an open-source NoSQL (non-relational) database management system
- It is a document-oriented database
- Designed to store, query, and manage large volumes of data in a flexible and scalable manner
- It is schema-less meaning that documents in a collection can have different fields and structures

What is PostgreSQL?

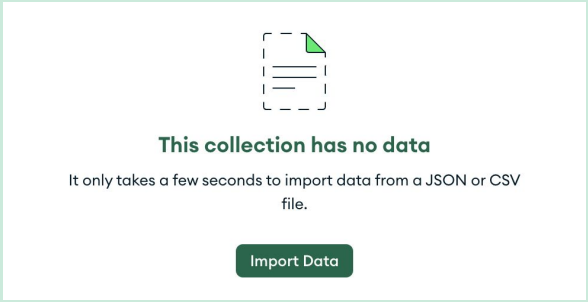
- It is an open-source Relational Database Management System **(RDBMS)**
- It supports SQL (Structured Query Language) for working with relational databases
- Highly scalable allowing clustering, partitioning and replication
- Supports a wide range of datatypes from arrays to JSON
- Supports foreign keys and constraints

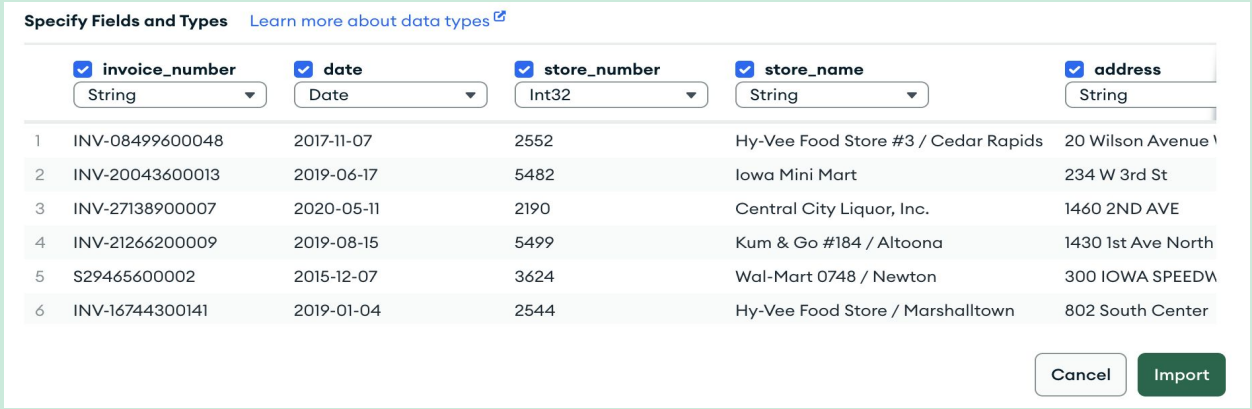
Dataset Overview

- Liquor sales in the US by county
- Contains over 5 million records with 23 columns
- Each row is an invoice record
- **Columns include information on:** invoice details, shops and vendors, specifications of items sold

Setting up the database in MongoDB

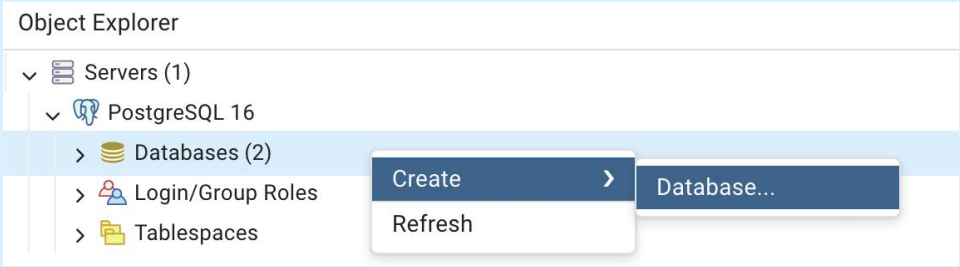
1) Manually create a DB in MongoDB using MongoDB Compass
2) Within the empty collection, click on ‘Import Data’ and choose the file from local computer
3) Click on import again in the following screen





Setting up the database in PostgreSQL

1) Manually create a DB in Postgres from the ‘Object Explorer’
2) Run the following python script to load the csv file from local computer to the postgres
3) Run the following Postgres command to alter the date column type from ‘text’ to ‘date’




```
acc_pool = pd.read_csv("/Users/pc_username/Documents/dw_project/liquor_sales.csv")

engine = create_engine('postgresql://postgres:postgres@localhost:5432/project')

acc_pool.to_sql('ls', engine ,if_exists='replace',index=False)
```

```
ALTER TABLE ls ALTER COLUMN "date" TYPE date
USING "date"::date
```

Query 1: What are the sales that are between 500 and 5000 USD?

MongoDB

```
roject> pipeline = [
..     {
..         $match: {
..             sale_usd: { $gte: 500, $lte: 5000 }
..         }
..     },
..     {
..         $project: {
..             invoice_number: 1,
..             date: 1,
..             store_name: 1,
..             sale_usd: 1,
..             _id: 0
..         }
..     },
..     {
..         $sort: {
..             sale_usd: -1
..         }
..     }
.. ];

{ '$match': { sale_usd: { '$gte': 500, '$lte': 5000 } } },
{
  '$project': { invoice_number: 1, date: 1, store_name: 1, sale_usd: 1, _id: 0 }
},
{ '$sort': { sale_usd: -1 } }
```

PostgreSQL

```
SELECT invoice_number, date,
store_name, sale_usd
FROM ls
WHERE sale_usd BETWEEN 500 AND 5000
ORDER BY sale_usd DESC;
```

Output 1: Sales between 500 and 5000 USD

MongoDB

```
[
[  {
     invoice_number: 'INV-11243200060',
     date: ISODate("2018-04-02T00:00:00.000Z"),
     store_name: 'Hy-Vee Food Store / Muscatine',
     sale_usd: 4998.6
   },
   {
     invoice_number: 'INV-11665200084',
     date: ISODate("2018-04-24T00:00:00.000Z"),
     store_name: 'Hy-Vee #7 / Cedar Rapids',
     sale_usd: 4998.6
   },
```

PostgreSQL

	invoice_number text	date date	store_name text	sale_usd double precision
1	INV-11274400002	2018-04-03	Fareway Stores #124 / Adel	4998.6
2	INV-08714100020	2017-11-16	Fareway Stores #153 / W Des Moines	4998.6
3	INV-05757700028	2017-06-27	Hy-Vee Food Store / Johnston	4998.6
4	INV-10620300060	2018-02-27	Hy-Vee Food Store / Johnston	4998.6
5	INV-08760500007	2017-11-20	Hy-Vee Wine and Spirits / Iowa City	4998.6

Query 2: Which years made the most profit?

MongoDB

```
project> pipeline = [
...     {
...         $group: {
...             _id: { $year: "$date" },
...             total_bottles_sold: { $sum: "$bottles_sold" },
...             total_sale_usd: { $sum: "$sale_usd" }
...         }
...     },
...     {
...         $sort: { total_sale_usd: -1 }
...     }
... ];
```

PostgreSQL

```
SELECT
    EXTRACT(YEAR FROM date) AS year,
    SUM(bottles_sold) AS total_bottles_sold,
    SUM(sale_usd) AS total_sale_usd
FROM ls
GROUP BY year
ORDER BY SUM(sale_usd) DESC;
```

Output 2: Which years made the most profit?

MongoDB


```
[
  {
    _id: 2019,
    total_bottles_sold: 6990974,
    total_sale_usd: 90567975.6
  },
  {
    _id: 2018,
    total_bottles_sold: 6547448,
    total_sale_usd: 86371077.89
  },
  {
    _id: 2017,
    total_bottles_sold: 6053476,
    total_sale_usd: 79207595.29
```


PostgreSQL

	year numeric	total_bottles_sold numeric	total_sale_usd double precision
1	2019	6990974	90567975.59999166
2	2018	6547448	86371077.88999465
3	2017	6053476	79207595.28999892
4	2020	5706871	74804342.1500005
5	2015	5571528	73752865.60000421

Query 3: Which stores sold more bottles than the average?

MongoDB

```
project> var avg_bottles_sold = db.ls.aggregate([
...     {
...         $group: {
...             _id: null,
...             avg_bottles_sold: { $avg: "$bottles_sold" }
...         }
...     }
... ]).next().avg_bottles_sold;

project> var result = db.ls.aggregate([
...     {
...         $match: {
...             bottles_sold: { $gt: avg_bottles_sold }
...         }
...     },
...     {
...         $project: {
...             store_name: 1,
...             address: 1,
...             city: 1,
...             county: 1,
...             bottles_sold: 1
...         }
...     }
... ]);
```

PostgreSQL

```
SELECT
      store_name, address,
      city, county,
      bottles_sold
FROM ls
WHERE bottles_sold >
       (SELECT AVG(bottles_sold) FROM ls)
```

Output 3: Stores with more bottle sales than the average

MongoDB

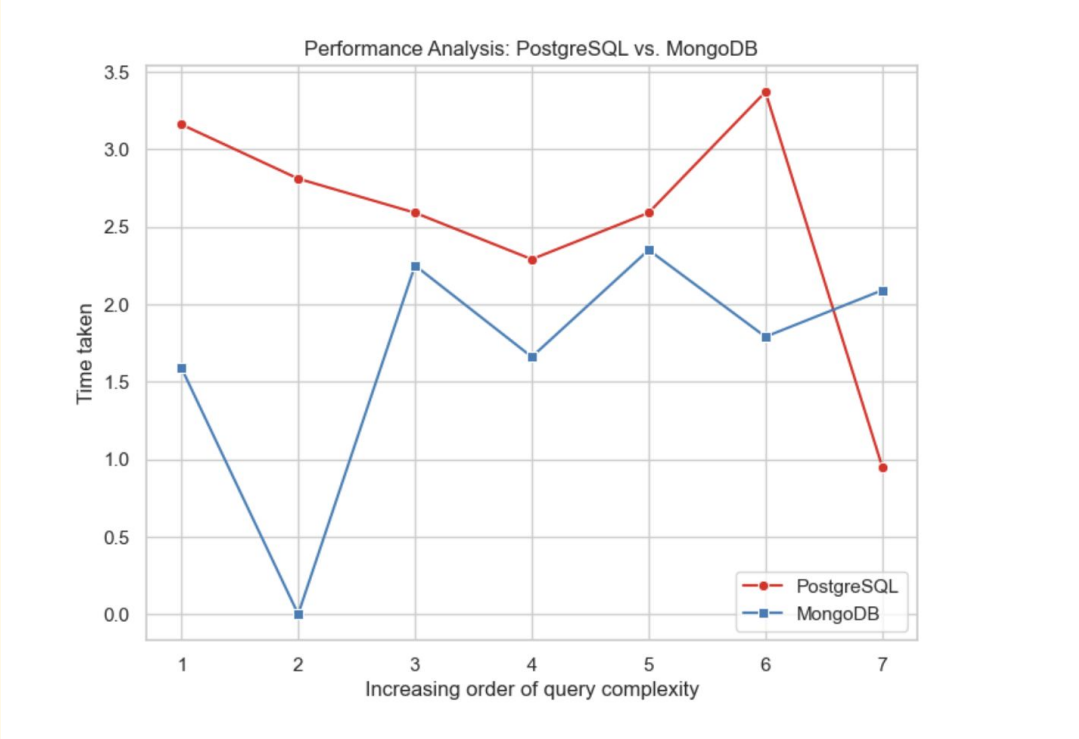
```
project> print(result)
[
  {
    _id: ObjectId("656bdf9121918ee2899b5a2d"),
    store_name: 'Sycamore Convenience',
    address: '617 Sycamore',
    city: 'Waterloo',
    county: 'BLACK HAWK',
    bottles_sold: 12
  },
  {
    _id: ObjectId("656bdf9121918ee2899b5a31"),
    store_name: 'Hy-Vee Food Store #1 / Council Bluffs',
    address: '2323 W Broadway',
    city: 'Council Bluffs',
    county: 'POTTAWATTA',
    bottles_sold: 12
  },
  {
    _id: ObjectId("656bdf9121918ee2899b5a33"),
    store_name: 'Benz Distributing',
    address: '501 7TH AVE SE',
    city: 'CEDAR RAPIDS',
    county: 'Linn',
    bottles_sold: 12
  },
```

PostgreSQL

	store_name text	address text	city text	county text	bottles_sold bigint
1	Sycamore Convenience	617 Sycamore	Waterloo	BLACK HAWK	12
2	Hy-Vee Food Store #1 / C...	2323 W Broadway	Council Bluffs	POTTAWATTA	12
3	Benz Distributing	501 7TH AVE SE	CEDAR RAPIDS	Linn	12
4	Barmuda Distribution	6027 University Ave St...	Cedar Falls	BLACK HAWK	12
5	Kum & Go #135 / Polk City	303 W Broadway St	Polk City	POLK	12
6	Casey's General Store #9...	113, NE 3RD ST	PANORA	Guthrie	12
7	Urbandale Liquor	6401 Douglas Ave STE 1	Urbandale	POLK	12
8	Hy-Vee Food Store #3 / C...	20 WILSON AVENUE W...	CEDAR RAPIDS	Linn	12
9	Sam's Club 6514 / Waterl...	210 East Tower Park Dr	Waterloo	BLACK HAWK	12
10	Hy-Vee Food Store / Fleu...	4605 Fleur Drive	Des Moines	POLK	12
11	Sam's Club 8238 / Daven...	3845 ELMORE AVE.	DAVENPORT	Scott	12
12	Washington Street Mini ...	1601 WASHINGTON ST	DAVENPORT	Scott	24
13	Kum & Go #4127 / SLOAN	1864 HWY 141	SLOAN	Woodbury	12
14	Kimberly Mart / Davenport	1714 E Kimberly Rd	Davenport	Scott	24
15	One Stop Shop #3 / Algo...	220 S Phillips St	Algona	KOSSUTH	48

Performance Analysis

PostgreSQL	MongoDB	% comparison
3.16	1.59	97.8%
2.81	0.006	significantly faster
2.59	2.25	15.07%
2.29	1.66	37.67%
2.59	2.35	10.04%.
3.37	1.79	87.75%
0.95	2.09	54.23% (slower)



Thank you!