NoSQL Data Architectures Project Report

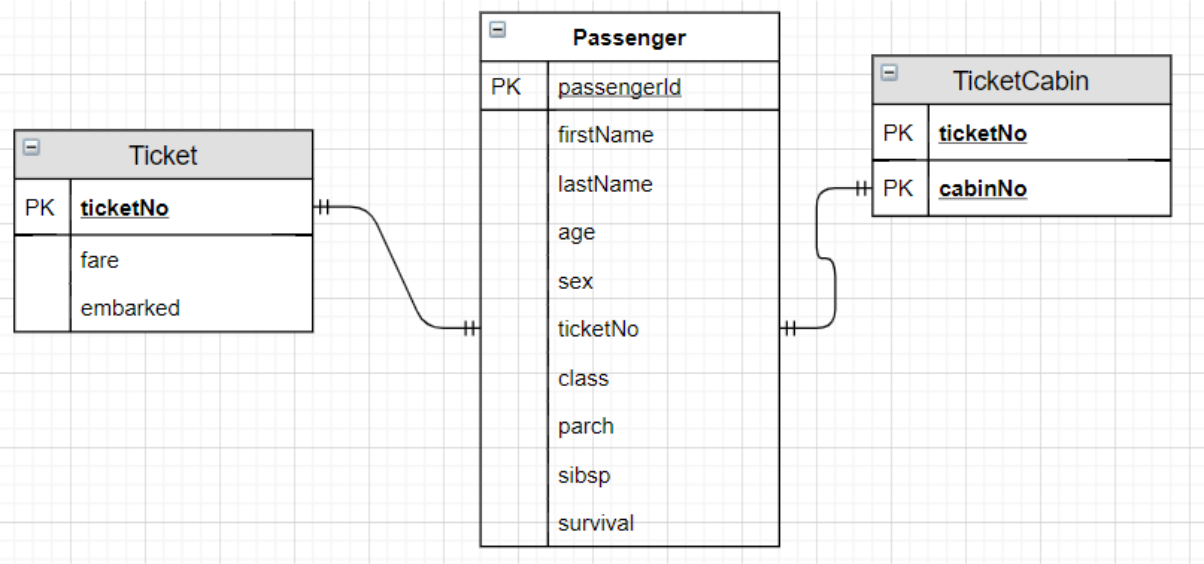
Roshan Sreekanth R00170592

roshan.sreekanth@mycit.ie

Contents

SQL Database

Database design



SQL CODE

```
CREATE TABLE ticket (
  `ticketNo` VARCHAR(255) NOT NULL,
  `fare` DECIMAL(18,4) NULL,
  `embarked` ENUM('C', 'Q', 'S', 'Unknown'),
  PRIMARY KEY (`ticketNo`));

CREATE TABLE passenger
(
		passengerId INT,
		firstName VARCHAR(255),
		lastName VARCHAR(255),
		age DECIMAL(18, 2),
		sex ENUM('male', 'female'),
		ticketNo VARCHAR(255),
		class ENUM ('1', '2', '3'),
```

```
            parch INT,
            sibsp INT,
            survival ENUM('Yes', 'No'),
            PRIMARY KEY (passengerId),
            FOREIGN KEY (ticketNo) REFERENCES ticket(ticketNo)
);

CREATE TABLE cabin
(
   cabinNo VARCHAR(255),
   ticketNo VARCHAR(255),
   PRIMARY KEY (cabinNo, ticketNo),
   FOREIGN KEY (ticketNo) REFERENCES ticket(ticketNo)
);
```

Insertion Code

```
import mysql.connector
import csv

class Passenger(object):
    def __init__(self, id, survival, socialStatus, lName, fName, sex, age, sib
ling_spouse, parent_child, ticketNo, fare, cabinNo, embarked):
        self.id = id
        self.survival = survival
        self.socialStatus = socialStatus
        self.lName = lName
        self.fName = fName
        self.sex = sex
        self.age = age
        self.sibling_spouse = sibling_spouse
        self.parent_child = parent_child
        self.ticketNo = ticketNo
        self.fare = fare
        self.cabinNo = cabinNo
        self.embarked = embarked

passengersList = []

with open('titanic-passengers3.csv', 'r') as file:
    reader = csv.reader(file)
    for index, row in enumerate(reader):
        if index != 0:
```

```
            id = int(row[0].strip())
            survival = row[1].strip()
            socialStatus = int(row[2].strip())
            lName = row[3].strip()
            fName = row[4].strip()
            sex = row[5].strip()
            age = -1
            if len(row[6]) != 0:
                age = float(row[6].strip())
            sibling_spouse = int(row[7].strip())
            parent_child = int(row[8].strip())
            ticketNo = row[9].strip()
            fare = float(row[10].strip())
            cabinNo = "None"
            if len(row[11]) != 0:
                cabinNo = row[11].strip()
            embarked = "Unknown"
            if len(row[12]) != 0:
                embarked = row[12].strip()

            passenger = Passenger(id, survival, socialStatus, lName, fName, se
x, age, sibling_spouse, parent_child, ticketNo, fare, cabinNo, embarked)
            passengersList.append(passenger)

mydb = mysql.connector.connect(
    host = "localhost",
    user = "root",
    password = "root",
    database = 'titanic',
    auth_plugin = 'mysql_native_password'
    )

unique_ticket = []
unique_cabinNo  = []
unique_ticketcabin = []
unique_composite = []

mycursor = mydb.cursor()
for passenger in passengersList:
    sql = "INSERT INTO ticket (ticketNo, fare, embarked) VALUES (%s, %s, %s)"
    val = (passenger.ticketNo, passenger.fare, passenger.embarked)
    if passenger.ticketNo not in unique_ticket:
        mycursor.execute(sql, val)

    unique_ticket.append(passenger.ticketNo)
```

```
    sql = "INSERT INTO passenger (passengerId, firstName, lastName, age, sex,
ticketNo, class, parch, sibsp, survival) VALUES (%s, %s, %s, %s, %s, %s, %s, %
s, %s, %s)"
    val = (passenger.id, passenger.fName, passenger.lName, passenger.age, pass
enger.sex, passenger.ticketNo, passenger.socialStatus, passenger.parent_child,
 passenger.sibling_spouse, passenger.survival)
    mycursor.execute(sql, val)

    sql = "INSERT INTO cabin (ticketNo, cabinNo) VALUES (%s, %s)"
    val = (passenger.ticketNo, passenger.cabinNo)
    if passenger.cabinNo != "None" and passenger.ticketNo + passenger.cabinNo
not in unique_composite:
        mycursor.execute(sql, val)
    unique_composite.append(passenger.ticketNo + passenger.cabinNo)

    mydb.commit()


mycursor.close()
mydb.close()
file.close()
```

Queries

1) **List alphabetically the names of all female passengers that survived**

   SELECT * FROM passenger WHERE sex = 'female' AND survival = 'Yes' ORDER BY lastName ASC;

passengerId	firstName	lastName	age	sex	ticketNo	class	parch	sibsp	survival
280	Mrs. Stanton (Rosa Hunt)	Abbott	35.00	female	C.A. 2673	3	1	1	Yes
875	Mrs. Samuel (Hannah Wizosky)	Abelson	28.00	female	P/PP 3381	2	0	1	Yes
856	Mrs. Sam (Leah Rosen)	Aks	18.00	female	392091	3	1	0	Yes
731	Miss. Elisabeth Walton	Allen	29.00	female	24160	1	0	0	Yes
193	Miss. Carla Christine Nielsine	Andersen-Jensen	19.00	female	350046	3	0	1	Yes
69	Miss. Erna Alexandra	Andersson	17.00	female	3101281	3	2	4	Yes
276	Miss. Kornelia Theodosia	Andrews	63.00	female	13502	1	0	1	Yes
519	Mrs. William A (Florence Mary Agnes Hughes)	Angle	36.00	female	226875	2	0	1	Yes
572	Mrs. Edward Dale (Charlotte Lamson)	Appleton	53.00	female	11769	1	0	2	Yes
26	Mrs. Carl Oscar (Selma Augusta Emilia Johansson)	Asplund	38.00	female	347077	3	5	1	Yes
234	Miss. Lillian Gertrud	Asplund	5.00	female	347077	3	2	4	Yes
701	Mrs. John Jacob (Madeleine Talmadge Force)	Astor	18.00	female	PC 17757	1	0	1	Yes
370	Mme. Leontine Pauline	Aubart	24.00	female	PC 17477	1	0	0	Yes
781	Miss. Banoura	Ayoub	13.00	female	2687	3	0	0	Yes
86	Mrs. Karl Alfred (Maria Mathilda Gustafsson)	Backstrom	33.00	female	3101278	3	0	3	Yes
470	Miss. Helene Barbara	Baclini	0.75	female	2666	3	1	2	Yes

   Count: 233

2) **Output by class, the names of children who survived**

   SELECT * FROM passenger WHERE age BETWEEN 0 AND 17 AND survival = 'Yes' ORDER BY class

passengerId	firstName	lastName	age	sex	ticketNo	class	parch	sibsp	survival
306	Master. Hudson Trevor	Allison	0.92	male	113781	1	2	1	Yes
308	Mrs. Victor de Satode (Maria Josefa Perez de S...	Penasco y Castellana	17.00	female	PC 17758	1	0	1	Yes
330	Miss. Jean Gertrude	Hippach	16.00	female	111361	1	1	0	Yes
436	Miss. Lucile Polk	Carter	14.00	female	113760	1	2	1	Yes
551	Mr. John Borland Jr	Thayer	17.00	male	17421	1	2	0	Yes
854	Miss. Mary Conover	Lines	16.00	female	PC 17592	1	1	0	Yes
782	Mrs. Albert Adrian (Vera Gillespie)	Dick	17.00	female	17474	1	0	1	Yes
446	Master. Washington	Dodge	4.00	male	33638	1	2	0	Yes
803	Master. William Thornton II	Carter	11.00	male	113760	1	2	1	Yes
505	Miss. Roberta	Maioni	16.00	female	110152	1	0	0	Yes
690	Miss. Georgette Alexandra	Madill	15.00	female	24160	1	1	0	Yes
10	Mrs. Nicholas (Adele Achem)	Nasser	14.00	female	237736	2	0	1	Yes
531	Miss. Phyllis May	Quick	2.00	female	26360	2	1	1	Yes
536	Miss. Eva Miriam	Hart	7.00	female	F.C.C. 1...	2	2	0	Yes
550	Master. John Morgan Jr	Davies	8.00	male	C.A. 33112	2	1	1	Yes
44	Miss. Simonne Marie Anne Andree	Laroche	3.00	female	SC/Paris ...	2	2	1	Yes

Count: 61

3) Output by class, the names of children who did not survive

SELECT * FROM passenger WHERE age BETWEEN 0 AND 17 AND survival = 'No' ORDER BY class

passengerId	firstName	lastName	age	sex	ticketNo	class	parch	sibsp	survival
298	Miss. Helen Loraine	Allison	2.00	female	113781	1	2	1	No
792	Mr. Alfred	Gaskell	16.00	male	239865	2	0	0	No
842	Mr. Thomas Charles	Mudd	16.00	male	S.O./P.P. 3	2	0	0	No
17	Master. Eugene	Rice	2.00	male	382652	3	1	4	No
25	Miss. Torborg Danira	Palsson	8.00	female	349909	3	1	3	No
51	Master. Juha Niilo	Panula	7.00	male	3101295	3	1	4	No
60	Master. William Frederick	Goodwin	11.00	male	CA 2144	3	2	5	No
64	Master. Harald	Skoog	4.00	male	347088	3	2	3	No
72	Miss. Lillian Amy	Goodwin	16.00	female	CA 2144	3	2	5	No
87	Mr. William Neal	Ford	16.00	male	W./C. 6608	3	3	1	No
112	Miss. Hileni	Zabour	14.50	female	2665	3	0	1	No
115	Miss. Malake	Attalah	17.00	female	2627	3	0	0	No
120	Miss. Ellis Anna Maria	Andersson	2.00	female	347082	3	2	4	No
139	Mr. Olaf Elon	Osen	16.00	male	7534	3	0	0	No
148	Miss. Robina Maggie Ruby	Ford	9.00	female	W./C. 6608	3	2	2	No
164	Mr. Jovo	Calic	17.00	male	315093	3	0	0	No

Count: 52

4) Count of children in each class that survived

SELECT class, COUNT(*) FROM passenger WHERE age BETWEEN 0 AND 17 AND survival = 'Yes' GROUP BY class

class	COUNT(*)
2	21
3	29
1	11

Count: 61

5) Count of children in each class who did not survive

SELECT class, COUNT(*) FROM passenger WHERE age BETWEEN 0 AND 17 AND survival = 'No' GROUP BY class

class	COUNT(*)
3	49
1	1
2	2

Count: 52

MongoDB Database

Importing Data from JSON

```
PS C:\Users\rosha> mongo
MongoDB shell version v3.4.24
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.24
Server has startup warnings:
2020-02-26T10:28:09.062+0000 I CONTROL  [initandlisten]
2020-02-26T10:28:09.062+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-02-26T10:28:09.063+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-02-26T10:28:09.063+0000 I CONTROL  [initandlisten]
> db
test
> db.createCollection("titanic")
{ "ok" : 1 }
> show dbs
admin  0.000GB
local  0.000GB
test   0.000GB
> show collections
titanic
> exit
bye
PS C:\Users\rosha>
```

```
PS C:\Users\rosha> mongoimport --db test --collection titanic --file "C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 1\titanic-passengers.json" --jsonArray
2020-02-26T10:32:40.614+0000    connected to: localhost
2020-02-26T10:32:40.630+0000    imported 891 documents
PS C:\Users\rosha>
```

Queries

1) **List alphabetically the names of all female passengers that survived**

db.titanic.find({"fields.sex" : "female", "fields.survived" : "Yes"}).sort({"fields.name" : 1}).pretty()


```
{
        "_id" : ObjectId("5e5649488f0f8847df66da6a"),
        "datasetid" : "titanic-passengers",
        "recordid" : "9746466badd5ffffcbceca2c13b61541caa39aab",
        "fields" : {
                "fare" : 20.25,
                "name" : "Abbott, Mrs. Stanton (Rosa Hunt)",
                "embarked" : "S",
                "age" : 35,
                "parch" : 1,
                "pclass" : 3,
                "sex" : "female",
                "survived" : "Yes",
                "ticket" : "C.A. 2673",
                "passengerid" : 280,
                "sibsp" : 1
        },
        "record_timestamp" : "2016-09-20T23:34:51.313+01:00"
}
{
        "_id" : ObjectId("5e5649488f0f8847df66d8f0"),
        "datasetid" : "titanic-passengers",
        "recordid" : "36d70190ba9a19447de634d789ec62ca747dc166",
        "fields" : {
                "fare" : 24,
                "name" : "Abelson, Mrs. Samuel (Hannah Wizosky)",
                "embarked" : "C",
                "age" : 28,
                "parch" : 0,
                "pclass" : 2,
                "sex" : "female",
                "survived" : "Yes",
                "ticket" : "P/PP 3381",
                "passengerid" : 875,
                "sibsp" : 1
        },
        "record_timestamp" : "2016-09-20T23:34:51.313+01:00"
}
{
        "_id" : ObjectId("5e5649488f0f8847df66db63"),
        "datasetid" : "titanic-passengers",
        "recordid" : "9d67a62ba4534eb6cb37057b9932cabb5298a43f",
        "fields" : {
                "fare" : 9.35,
                "name" : "Aks, Mrs. Sam (Leah Rosen)",
                "embarked" : "S",
                "age" : 18,
                "parch" : 1,
                "pclass" : 3,
                "sex" : "female",
                "survived" : "Yes",
                "ticket" : "392091",
                "passengerid" : 856,
                "sibsp" : 0
        },
```


Count: 233

2) Output by class, the names of children who survived

db.titanic.find({"fields.age" : {$gte : 0}, "fields.age" : {$lte : 17}, "fields.survived" : "Yes"}).sort({"fields.pclass" : 1}).pretty()


```
{
        "_id" : ObjectId("5e5649488f0f8847df66d94d"),
        "datasetid" : "titanic-passengers",
        "recordid" : "b25e6694eed2755983b67f6d84702e209fdb16f4",
        "fields" : {
                "fare" : 39.4,
                "name" : "Lines, Miss. Mary Conover",
                "embarked" : "S",
                "age" : 16,
                "parch" : 1,
                "pclass" : 1,
                "sex" : "female",
                "survived" : "Yes",
                "sibsp" : 0,
                "passengerid" : 854,
                "ticket" : "PC 17592",
                "cabin" : "D28"
        },
        "record_timestamp" : "2016-09-20T23:34:51.313+01:00"
}
{
        "_id" : ObjectId("5e5649488f0f8847df66d99b"),
        "datasetid" : "titanic-passengers",
        "recordid" : "3f99c6b822731c683080651cc4015c8660bd0a71",
        "fields" : {
                "fare" : 81.8583,
                "name" : "Dodge, Master. Washington",
                "embarked" : "S",
                "age" : 4,
                "parch" : 2,
                "pclass" : 1,
                "sex" : "male",
                "survived" : "Yes",
                "sibsp" : 0,
                "passengerid" : 446,
                "ticket" : "33638",
                "cabin" : "A34"
        },
        "record_timestamp" : "2016-09-20T23:34:51.313+01:00"
}
{
        "_id" : ObjectId("5e5649488f0f8847df66d9d3"),
        "datasetid" : "titanic-passengers",
        "recordid" : "3daa488a6b10d539f7f4e2f511b8bb4b4e652d3e",
        "fields" : {
                "fare" : 86.5,
                "name" : "Maioni, Miss. Roberta",
                "embarked" : "S",
                "age" : 16,
```


Count: 61

3) **Output by class, the names of children who did not survive**

db.titanic.find({"fields.age" : {$gte : 0}, "fields.age" : {$lte : 17}, "fields.survived" : "No"}).sort({"fields.pclass" : 1}).pretty()


```
{
        "_id" : ObjectId("5e5649488f0f8847df66d9e7"),
        "datasetid" : "titanic-passengers",
        "recordid" : "218ca6342a742648a74991523a0bd0970c13853a",
        "fields" : {
                "fare" : 151.55,
                "name" : "Allison, Miss. Helen Loraine",
                "embarked" : "S",
                "age" : 2,
                "parch" : 2,
                "pclass" : 1,
                "sex" : "female",
                "survived" : "No",
                "sibsp" : 1,
                "passengerid" : 298,
                "ticket" : "113781",
                "cabin" : "C22 C26"
        },
        "record_timestamp" : "2016-09-20T23:34:51.313+01:00"
}
{
        "_id" : ObjectId("5e5649488f0f8847df66d92a"),
        "datasetid" : "titanic-passengers",
        "recordid" : "159b6c4aaaf598867425acaa6801611bfa5c6c7d",
        "fields" : {
                "fare" : 10.5,
                "name" : "Mudd, Mr. Thomas Charles",
                "embarked" : "S",
                "age" : 16,
                "parch" : 0,
                "pclass" : 2,
                "sex" : "male",
                "survived" : "No",
                "ticket" : "S.O./P.P. 3",
                "passengerid" : 842,
                "sibsp" : 0
        },
        "record_timestamp" : "2016-09-20T23:34:51.313+01:00"
}
{
        "_id" : ObjectId("5e5649488f0f8847df66d9c3"),
        "datasetid" : "titanic-passengers",
        "recordid" : "da6113c1de345e748d0424885b38998340e3ede0",
        "fields" : {
                "fare" : 26,
                "name" : "Gaskell, Mr. Alfred",
                "embarked" : "S",
                "age" : 16,
                "parch" : 0,
                "pclass" : 2,
                "sex" : "male",
                "survived" : "No",
                "ticket" : "239865",
```


Count: 52

4) **Count of children in each class that survived**

```
db.titanic.aggregate([
        {
                $match : {"fields.age" : {$gte : 0}, "fields.age" : {$lte : 17},
                "fields.survived" : "Yes"}
        },
        {
                $group :
                {
                        _id : "$fields.pclass",
                        count : {$sum : 1}
                }
        }
])
```

```
{ "_id" : 1, "count" : 11 }
{ "_id" : 2, "count" : 21 }
{ "_id" : 3, "count" : 29 }
```

Count: 61

5) **Count of children in each class who did not survive**

```
db.titanic.aggregate([
        {
                $match : {"fields.age" : {$gte : 0}, "fields.age" : {$lte : 17},
                "fields.survived" : "No"}
        },
        {
                $group :
                {
                        _id : "$fields.pclass",
                        count : {$sum : 1}
                }
        }
])
```

```
{ "_id" : 1, "count" : 1 }
{ "_id" : 2, "count" : 2 }
{ "_id" : 3, "count" : 49 }
```

Count: 52

Neo4j

Importing data

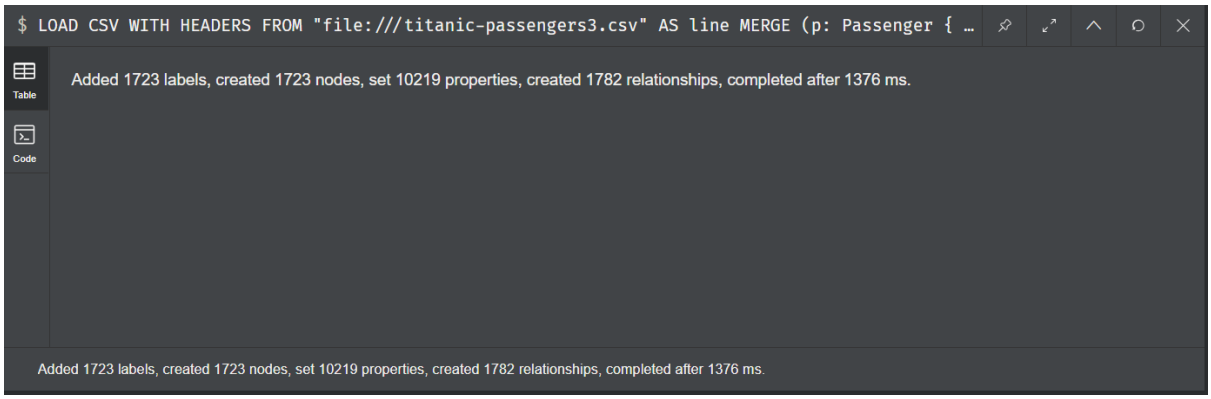
```
$ LOAD CSV FROM "file:///titanic-passengers3.csv" AS passengers RETURN passengers
```

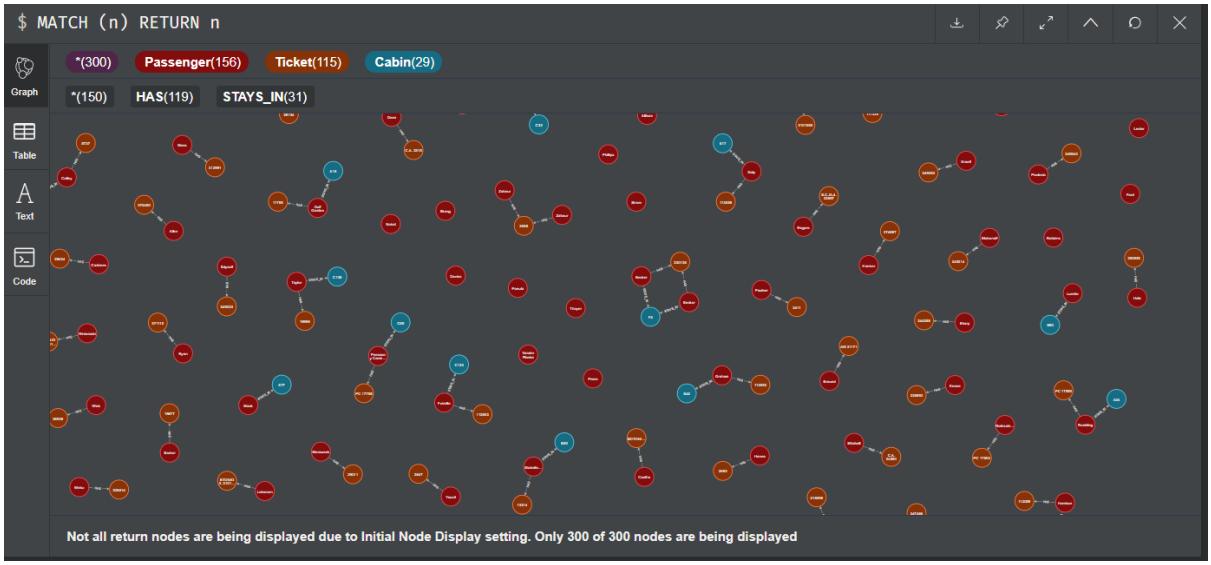
Creating Graph and Relationships

```
LOAD CSV WITH HEADERS FROM "file:///titanic-passengers3.csv" AS line
MERGE (p: Passenger
        {
          id: toInteger(line.PassengerId),
          survived: line.Survived,
          class: toInteger(coalesce(line.Pclass, 0)),
          lastName: line.`Last Name`,
          firstName: line.Firstname,
          sex: coalesce(line.Sex, "Unknown"),
          age: toFloat(coalesce(line.Age, -1)),
          sibSp: line.SibSp,
          parch: line.Parch
        }
      )
```

```
MERGE (t: Ticket
        {
            ticketNo: line.Ticket,
            fare: toFloat(line.Fare),
            embarked: coalesce(line.Embarked, "Unknown")
        }
      )

MERGE (c: Cabin
        {
            cabinNo: coalesce(line.Cabin, "Unknown")
        }
      )
CREATE (p)-[:HAS]->(t)
CREATE (p)-[:STAYS_IN]->(c)
```

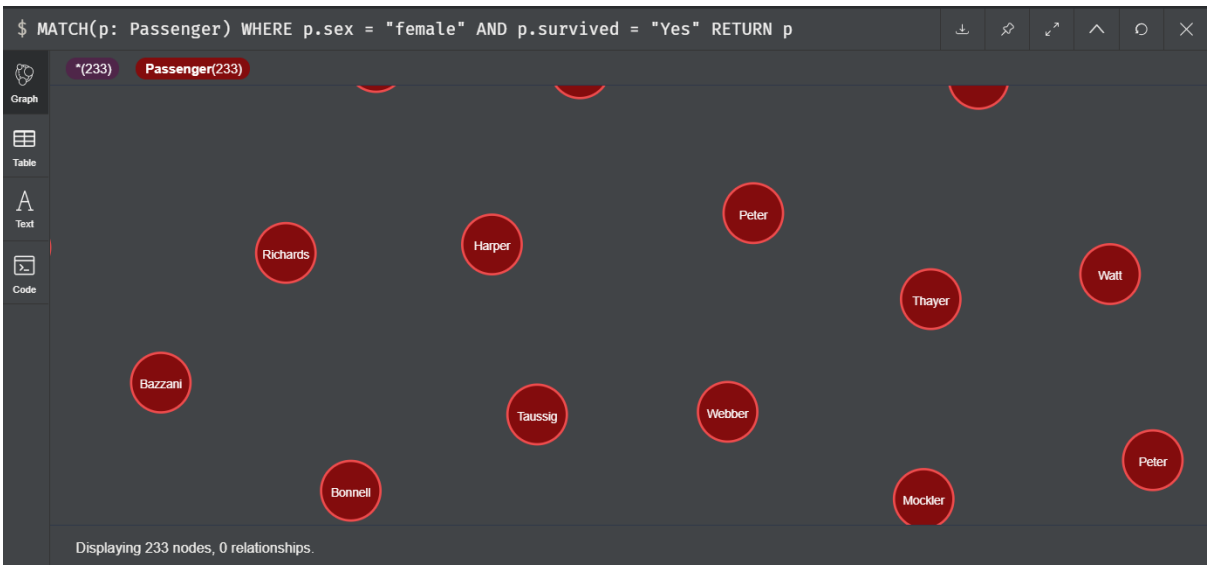




Queries

1) **List alphabetically the names of all female passengers that survived**

```
MATCH(p: Passenger)
WHERE p.sex = "female" AND p.survived = "Yes"
 RETURN p
```



Count: 233

2) **Output by class, the names of children who survived**

```
MATCH(p: Passenger)
WHERE p.age >=0 AND p.age <= 17 AND p.survived = "Yes"
RETURN p
ORDER BY p.class
```


```
{"firstName":" Master. Hudson Trevor","lastName":"Allison","sex":"male","id":306,"survived":"Yes","sibSp":"1","class":1,"parch":"2","age":0.92}
{"firstName":" Miss. Mary Conover","lastName":"Lines","sex":"female","id":854,"survived":"Yes","sibSp":"0","class":1,"parch":"1","age":16.0}
{"firstName":" Master. Washington","lastName":"Dodge","sex":"male","id":446,"survived":"Yes","sibSp":"0","class":1,"parch":"2","age":4.0}
{"firstName":" Master. Edmond Roger","lastName":"Navratil","sex":"male","id":341,"survived":"Yes","sibSp":"1","class":2,"parch":"1","age":2.0}
{"lastName":"Ilett","firstName":" Miss. Bertha","sex":"female","id":85,"survived":"Yes","sibSp":"0","class":2,"age":17.0,"parch":"0"}
{"lastName":"West","firstName":" Miss. Constance Mirium","sex":"female","id":59,"survived":"Yes","sibSp":"1","class":2,"age":5.0,"parch":"2"}
{"firstName":" Master. John Morgan Jr","lastName":"Davies","sex":"male","id":550,"survived":"Yes","sibSp":"1","class":2,"parch":"1","age":8.0}
```


Count: 61

3) **Output by class, the names of children who did not survive**

```
MATCH(p: Passenger)
WHERE p.age >=0 AND p.age <= 17 AND p.survived = "No"
RETURN p
ORDER BY p.class
```

```
{"firstName":" Miss. Helen Loraine","lastName":"Allison","sex":"female
","id":298,"survived":"No","sibSp":"1","class":1,"parch":"2","age":2.0
}
```

```
{"lastName":"Mudd","firstName":" Mr. Thomas Charles","sex":"male","id"
:842,"survived":"No","sibSp":"0","class":2,"age":16.0,"parch":"0"}
```

```
{"lastName":"Gaskell","firstName":" Mr. Alfred","sex":"male","id":792,
"survived":"No","sibSp":"0","class":2,"age":16.0,"parch":"0"}
```

```
{"lastName":"Zabour","firstName":" Miss. Hileni","sex":"female","id":1
12,"survived":"No","sibSp":"1","class":3,"age":14.5,"parch":"0"}
```

```
{"firstName":" Miss. Nourelain","lastName":"Boulos","sex":"female","id
":853,"survived":"No","sibSp":"1","class":3,"parch":"1","age":9.0}
```

```
{"firstName":" Master. Harald","lastName":"Skoog","sex":"male","id":64
,"survived":"No","sibSp":"3","class":3,"parch":"2","age":4.0}
```

```
{"firstName":" Master. Karl Thorsten","lastName":"Skoog","sex":"male",
"id":820,"survived":"No","sibSp":"3","class":3,"parch":"2","age":10.0}
```


Count: 52

4) **Count of children in each class that survived**

MATCH(p: Passenger)

WHERE p.age >=0 AND p.age <= 17 AND p.survived = "Yes"

RETURN p.class AS `Class`, COUNT(p) AS `Children Survived` ORDER BY p.class

"Class"	"Children Survived"
1	11
2	21
3	29

5) **Count of children in each class who did not survive**

```
MATCH(p: Passenger)
WHERE p.age >=0 AND p.age <= 17 AND p.survived = "No"
RETURN p.class AS `Class`, COUNT(p) AS `Children Survived` ORDER BY p.class
```

"Class"	"Children Survived"
1	1
2	2
3	49

Conclusion

Each database had its own advantages and disadvantages. SQL had a detailed structure of Primary and Foreign keys, which made the process of creating tables easier. However, it's rigid enforcement of constraints makes it hard to modify the table after it has been constructed.

MongoDB's schema-less structure makes it easy to import ad directly work with data, making it more convenient (especially for JSON files). Queries are relatively harder to write as the syntax is more complicated than SQL.

Neo4j was the easiest to construct as it uses a natural graph relationship structure that is easy to visualize. The visualization of data is also a very useful and interactive feature. However, sharding is not possible, and so all the data must be stored on one machine, hence slowing it down when a large amount of data is queried.