Final Project Report

# Project title

Nobel Peace

Name of the project

Nobel Peace Prize Data Mangement

Team members

Atreya Raorane

Bhargavi

Professor

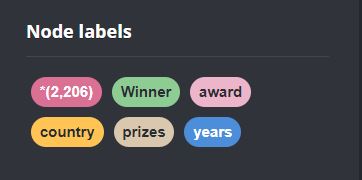
Prof. Kaleema

Project and Scope of Planning:

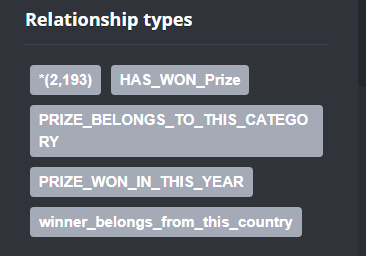
As the name suggest that the main purpose of the project is to keep all records of Nobel prizes won by people in different fields like peace, science, arts , politics etc. We are planning to make a working graph database that keep records of every data that is available about the Nobel prizes and relate the data in different nodes via relationships. We want to make a graph database with the help of neo4j database software to make storing the data very easy and fast to recover. We want make a reliable and consistent database that wont have problems like residency and inconsistency. Hence the making the database Significantly unique.

The main goal is to integrate all the data which is available in multiple csv file format into one single database. There are currently multiple csv files like winner, prizes, category of prizes, years, country. We intend to calculate number of prizes per person, number of prizes won by country, average age of person who has won prize, number of people who won the prize according to country etc. This can be done by integrating the csv files and then querying the database to get intended results. With this we can customize the database according to out needs.

**Different nodes in the database**:

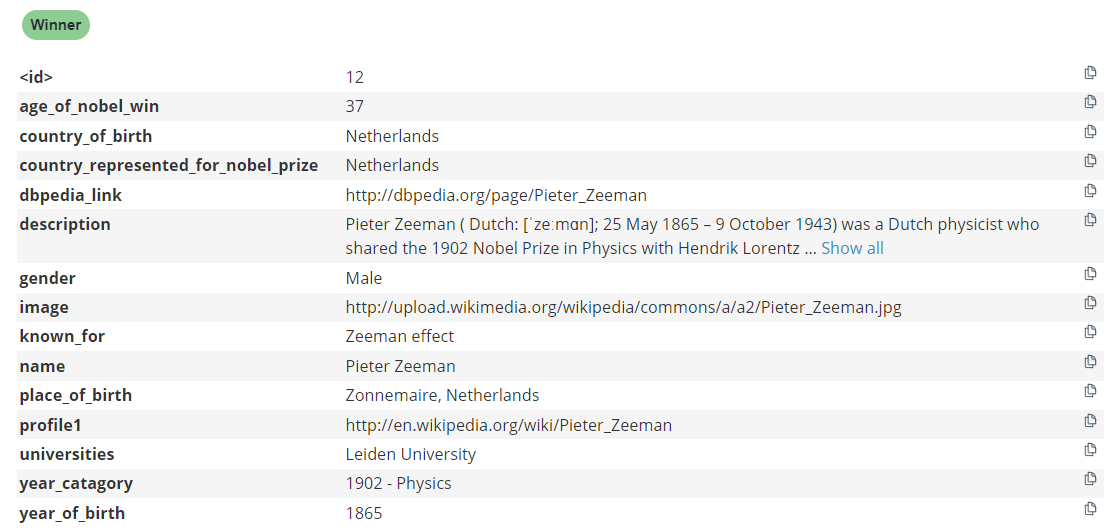


**Different Relationships in the Database:**

****

**Properties of different Types of nodes:**

**Winner Node:**

****

**Prizes Node:**

**A screenshot of a computer

Description automatically generated with low confidence**

**Country Node:**

****

**Award Node:**

****

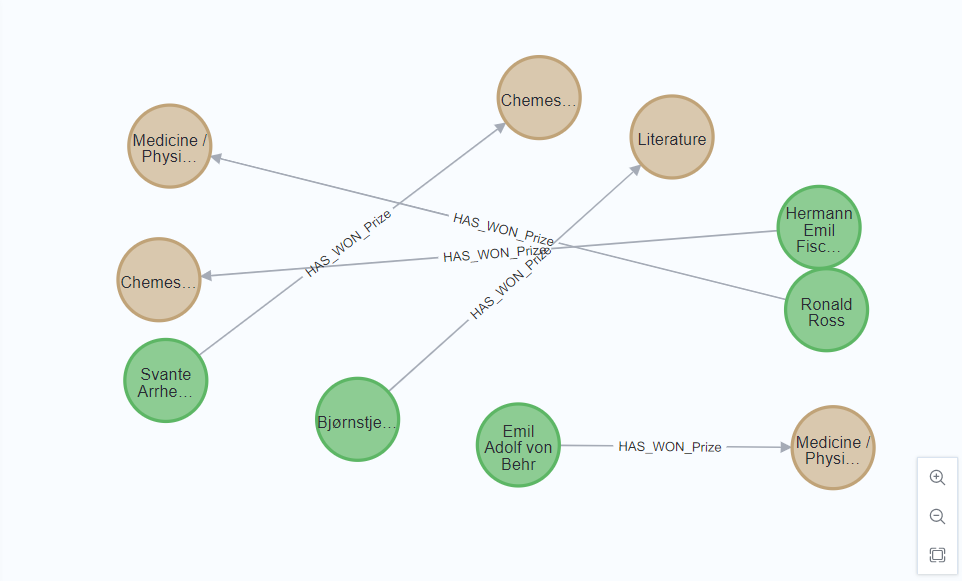
**Year Node:**

**A picture containing text, screenshot, font, number

Description automatically generated**

**Relationships of Between all Nodes:**

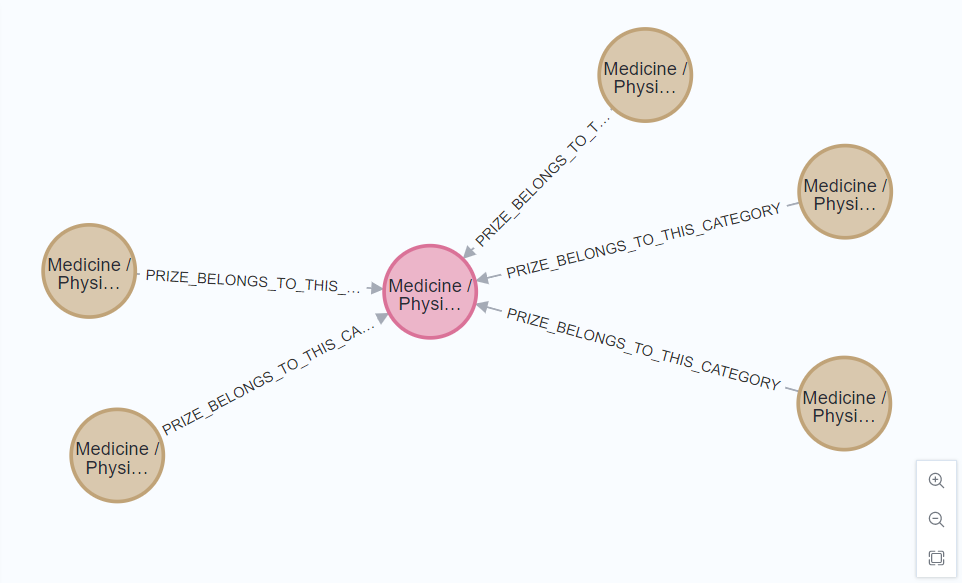
**HAS\_WON\_PRIZE:**

****

This relationship show the relation ship between 2 types of node that are Winner and Prize.

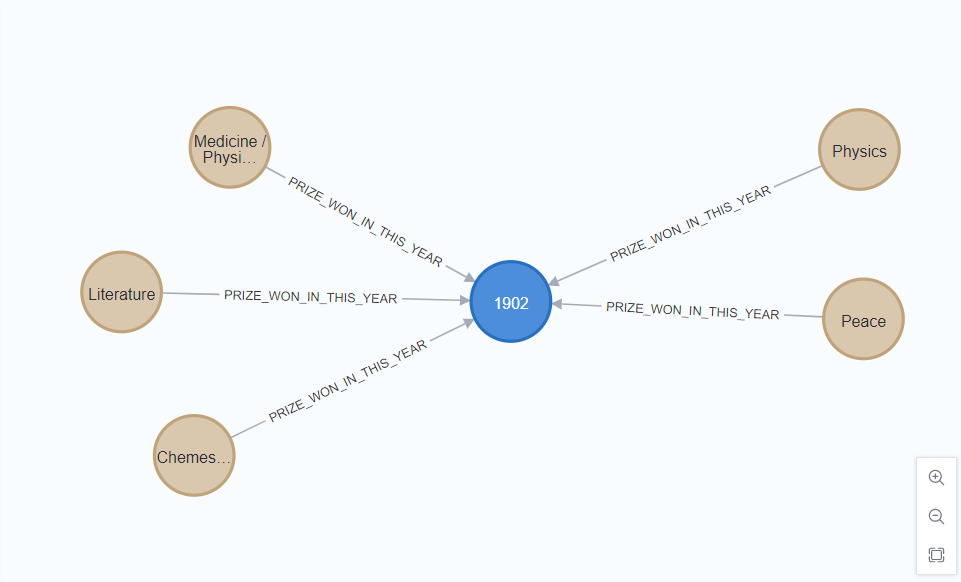
The above relation that this Winner has won this prize.

**PRIZE\_BELONGS\_TO\_THIS\_CATEGORY:**



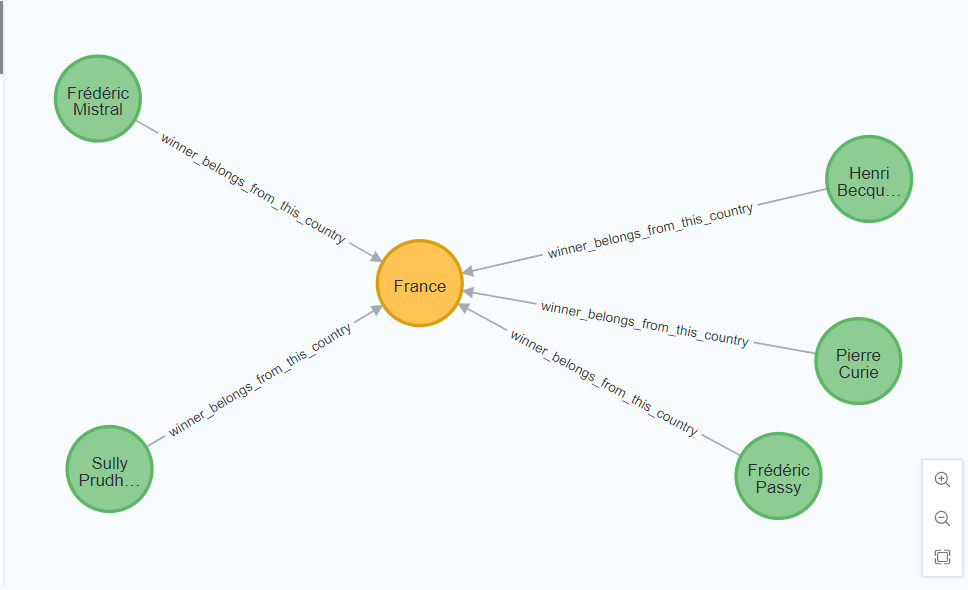
This relationship shows the relation between 2 nodes. Those nodes are award node and Prize node. This relationship shows that this prize belongs to this category. Award node depicts the category and Prize node depicts the nature of the prize like Peace-1902, medical-Research etc.

**PRIZE\_WON\_IN\_THIS\_YEAR:**



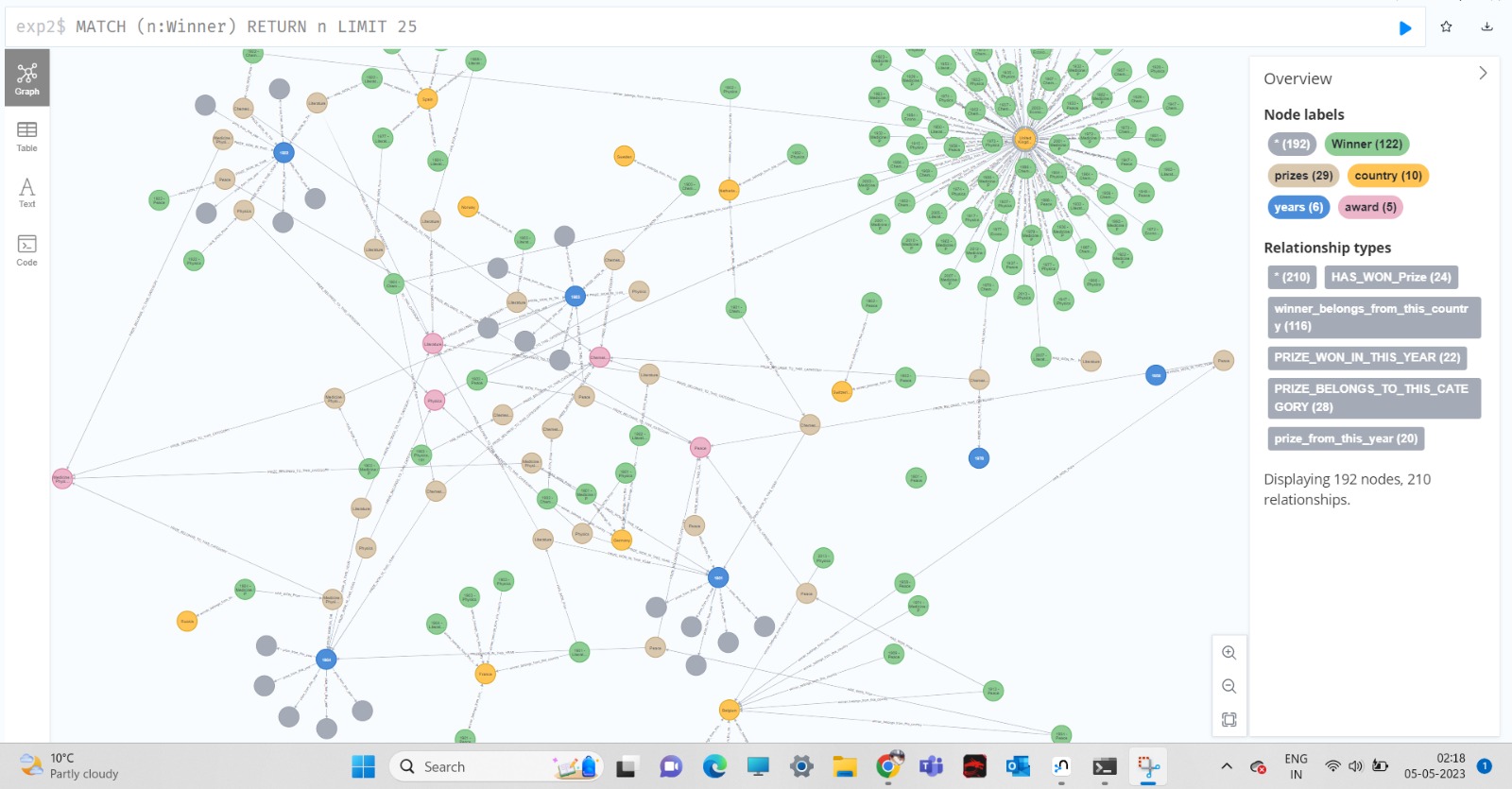
This relationship shows relationship between 2 nodes. Those are prizes and year. Here the relationship shows that the prize was won in this year. A particular prize belongs to this specific year. E.g. this Chemistry, literature, Peace etc all these prizes were won in 1902 year.

**Winner\_belongs\_from\_this\_country:**

****

The above relationships shows relationship between 2 nodes Winner and Country. This relations show that the particular winner belongs to this country.

**Data Model of the Graph Database:**

****

**Cipher Script to create the Database:**

CREATE

(n:WINNER{name:"Jacobus Henricus van 't Hoff", year\_catagory:"1901 - Chemestry", known\_for: "", field:"", gender: "Male",place\_of\_birth:"Rotterdam, Netherlands",country\_of\_birth:"Netherlands",short\_description:"Chemist",description:" Jacobus Henricus van 't Hoff, Jr. (30 August 1852 â€“ 1 March 1911) was a Dutch physical and organic chemist and the first winner of the Nobel Prize in Chemistry. He is best known for his discoveries in chemical kinetics, chemical equilibrium, osmotic pressure, and stereochemistry. Van 't Hoff's work in these subjects helped found the discipline of physical chemistry as it is today ",year\_of\_birth:1852,age\_of\_nobel\_win:49,universities:"",country\_represented\_for\_nobel\_prize:"Netherlands",image:"http://upload.wikimedia.org/wikipedia/commons/a/a4/Vant\_Hoff.jpg",dbpedia\_link:"http://dbpedia.org/page/Jacobus\_Henricus\_van\_%27t\_Hoff",profile:"http://en.wikipedia.org/wiki/Jacobus\_Henricus\_van\_%27t\_Hoff"})

CREATE

(p:prizes{id:"1901 - Chemestry",year:1901, known\_for: "",award:"Chemestry",winner: "Jacobus Henricus van 't Hoff",award\_motivation:"in recognition of the extraordinary services he has rendered by the discovery of the laws of chemical dynamics and osmotic pressure in solutions"

,link\_on\_official\_website:"http://www.nobelprize.org/nobel\_prizes/medicine/laureates/1901/"})

match (n:WINNER),(p:prizes) where n.name="Jacobus Henricus van 't Hoff" and p.winner ="Jacobus Henricus van 't Hoff" create(n)-[won:won\_the\_prize]->(p)

LOAD CSV WITH HEADERS FROM 'file:///winners.csv' AS row

CREATE (n:Winner{name: row.name, year\_catagory:row.http\_upload\_wikimedia\_org\_wikipedia\_commons\_6\_6e\_kailash\_satyarthi\_jpg

,known\_for:row.known\_for, field:row.field, gender:row.gender,place\_of\_birth:row.place\_of\_birth,country\_of\_birth:row.country\_of\_birth,short\_description:row.short\_description,description:row.description,year\_of\_birth:toInteger(row.year\_of\_birth),age\_of\_nobel\_win:toInteger(row.age\_of\_nobel\_win),universities:row.universities,country\_represented\_for\_nobel\_prize:row.country\_represented\_for\_nobel\_prize,image:row.image,dbpedia\_link:row.dbpedia\_link,profile1:row.profile})

LOAD CSV WITH HEADERS FROM 'file:///prizes.csv' AS row

CREATE (p:prizes{id: row.id,

year:toInteger(row.year),

award:row.award,

winner:row.winner,

award\_motivation:row.award\_motivation

})

MATCH (n:Winner), (p:prizes)

WHERE n.name = p.winner

MERGE (n)-[:HAS\_WON\_Prize]-(p)

LOAD CSV WITH HEADERS FROM 'file:///years.csv' AS row

CREATE (y1:years{year:toInteger(row.year),

average\_age\_of\_laureates:toInteger(row.average\_age\_of\_laureates),

prizes\_won\_by\_women:toInteger(row.prizes\_won\_by\_women), prizes\_won\_by\_men:toInteger(row.prizes\_won\_by\_men), percentage\_prizes\_won\_by\_women:toInteger(row.percentage\_prizes\_won\_by\_women),

percentage\_prizes\_won\_by\_men:toInteger(row.percentage\_prizes\_won\_by\_men),

remove\_average\_female\_winners\_1901\_2014:toInteger(row.remove\_average\_female\_winners\_1901\_2014),

remove\_average\_female\_winners\_1901\_2014\_2:toInteger(row.remove\_average\_female\_winners\_1901\_2014\_2)

})

LOAD CSV WITH HEADERS FROM 'file:///countries.csv' AS row

CREATE (c:country{country:row.country,

number\_of\_prizes:toInteger(row.number\_of\_prizes),

average\_age\_of\_laureates:toInteger(row.average\_age\_of\_laureates), prizes\_won\_by\_women:toInteger(row.prizes\_won\_by\_women), prizes\_won\_by\_men:toInteger(row.prizes\_won\_by\_men),

percentage\_prizes\_won\_by\_women:toInteger(row.percentage\_prizes\_won\_by\_women),

percentage\_prizes\_won\_by\_men:toInteger(row.percentage\_prizes\_won\_by\_men),

remove\_average\_female\_winners\_all\_countries:toInteger(row.remove\_average\_female\_winners\_all\_countries),

remove\_average\_female\_winners\_all\_countries\_2:toInteger(row.remove\_average\_female\_winners\_all\_countries\_2)

})

LOAD CSV WITH HEADERS FROM 'file:///award\_categories.csv' AS row

CREATE (a:award{category:row.category,

average\_age\_of\_laureates:toInteger(row.average\_age\_of\_laureates),

prizes\_won\_by\_women:toInteger(row.prizes\_won\_by\_women), prizes\_won\_by\_men:toInteger(row.prizes\_won\_by\_men), percentage\_prizes\_won\_by\_women:toInteger(row.percentage\_prizes\_won\_by\_women),

percentage\_prizes\_won\_by\_men:toInteger(row.percentage\_prizes\_won\_by\_men),

remove\_average\_female\_winners\_all\_countries:toInteger(row.remove\_average\_female\_winners\_all\_countries),

remove\_average\_female\_winners\_all\_countries:toInteger(row.remove\_average\_female\_winners\_all\_countries),

remove\_average\_female\_winners\_all\_countries\_2:toInteger(row.remove\_average\_female\_winners\_all\_countries\_2)

})

MATCH ()-[r]->()

RETURN type(r) as RelationshipType, count(r) as Count

ORDER BY Count DESC

MATCH (n:Winner)

WHERE n.age\_of\_nobel\_win > 40

RETURN n.name, n.age\_of\_nobel\_win

MATCH (n:Winner)-[:HAS\_WON\_Prize]->(p:prizes)

WITH n, count(p) AS num\_friends

WHERE num\_friends > 10

RETURN n.name, num\_friends

ORDER BY num\_friends DESC

LIMIT 10

MATCH (n:Winner)

RETURN SUM(n.age\_of\_nobel\_win) AS total\_age, AVG(n.age\_of\_nobel\_win) AS avg\_age, MIN(n.age\_of\_nobel\_win) AS min\_age, MAX(n.age\_of\_nobel\_win) AS max\_age

MATCH (p:prizes)

RETURN SUM(p.price) AS total\_price, AVG(p.price) AS avg\_price, MIN(p.price) AS min\_price, MAX(p.price) AS max\_price

MATCH (a:award), (p:prizes)

WHERE a.category = p.award

MERGE (p)-[:PRIZE\_BELONGS\_TO\_THIS\_CATEGORY]-(a)

MATCH (y1:year), (p:prizes)

WHERE y1.year = p.year

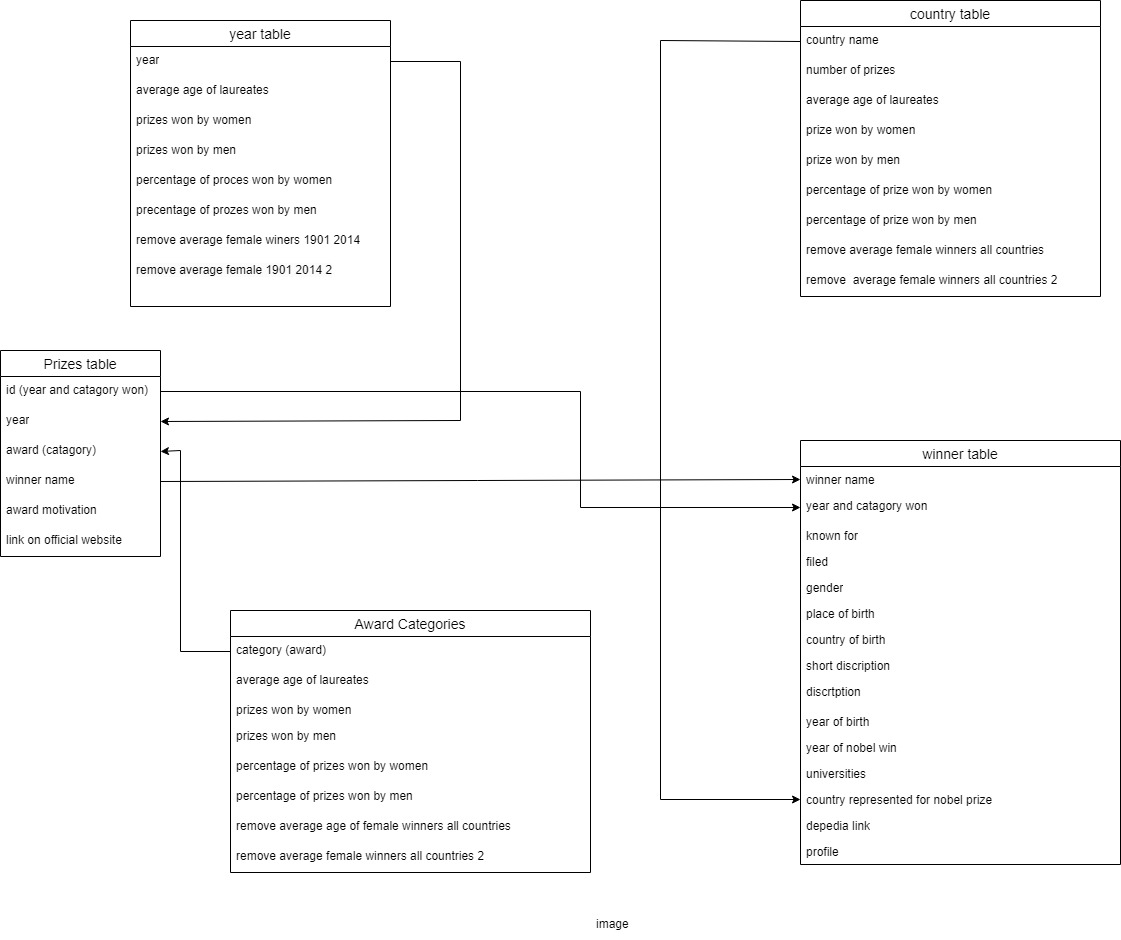
MERGE (p)-[:PRIZE\_WON\_IN\_THIS\_YEAR]-(y1)

MATCH (n:Winner), (c:country)

WHERE n.country\_represented\_for\_noble\_prize = c.country\_name

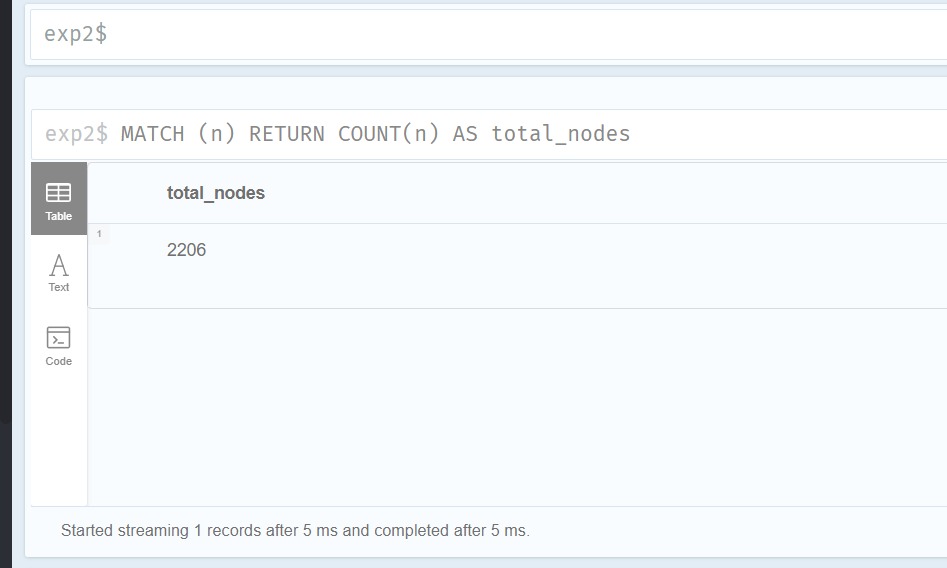
MERGE (a)-[:PRIZE\_BELONGS\_TO\_THIS\_CATEGORY]-(p)

**ER Diagram:**

****

**Different Operation Performed on The DATABASE:**

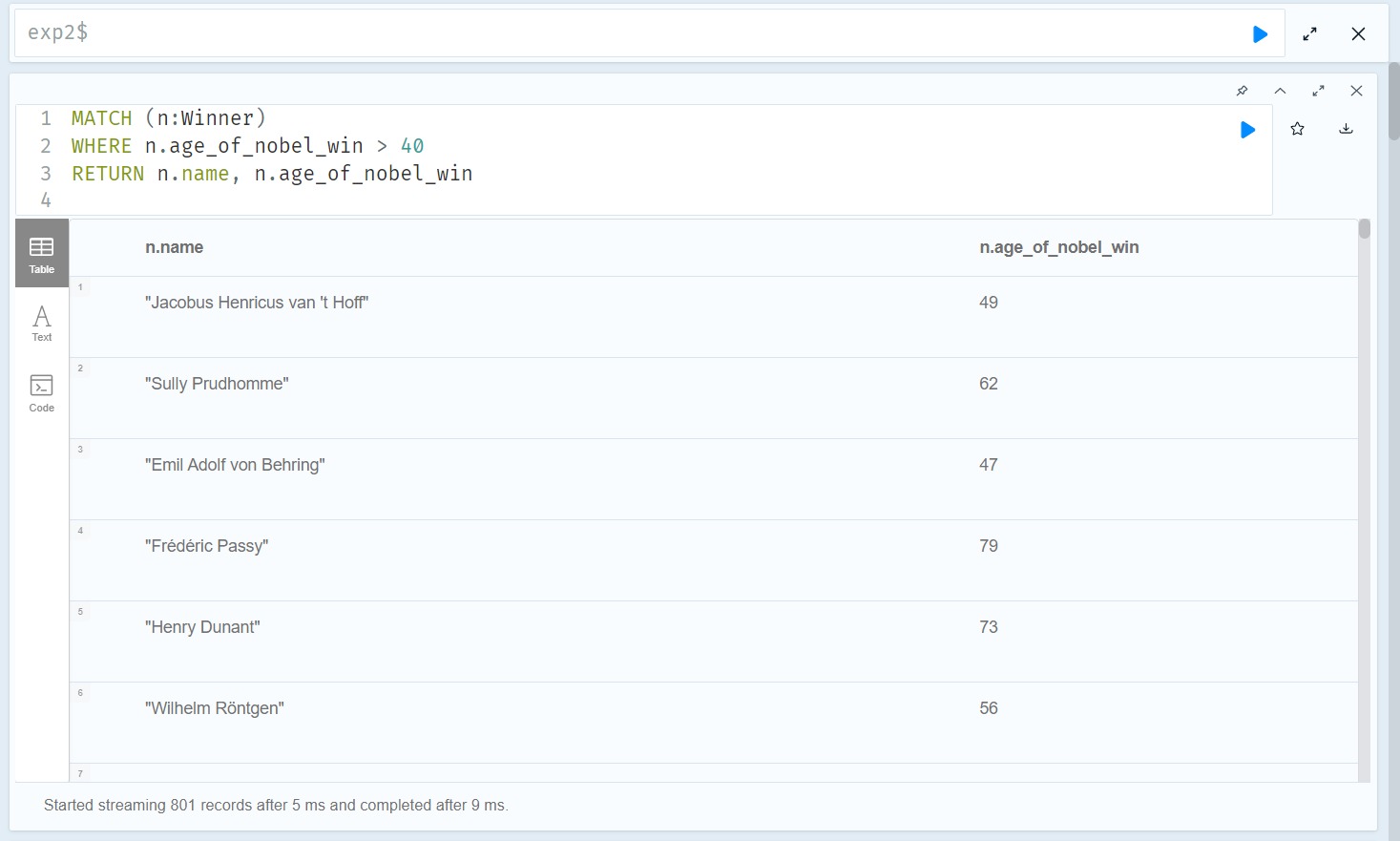
Display Number of Nodes in Database:



Performing sum, min, max, average operations on Winner Node to calculate age of winner.



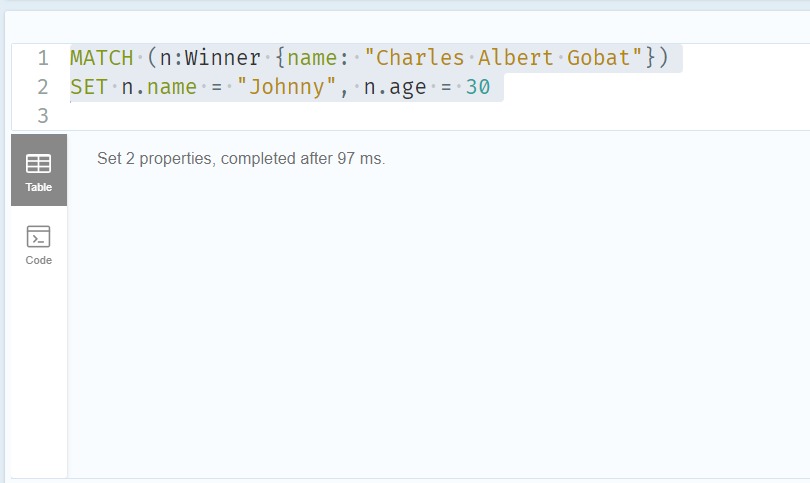
Performing Filtering Operation on The Database. Display Winner name and age if age is greater than 40.



Displaying total number of relationships in the Database.



Editing the data inside the node.

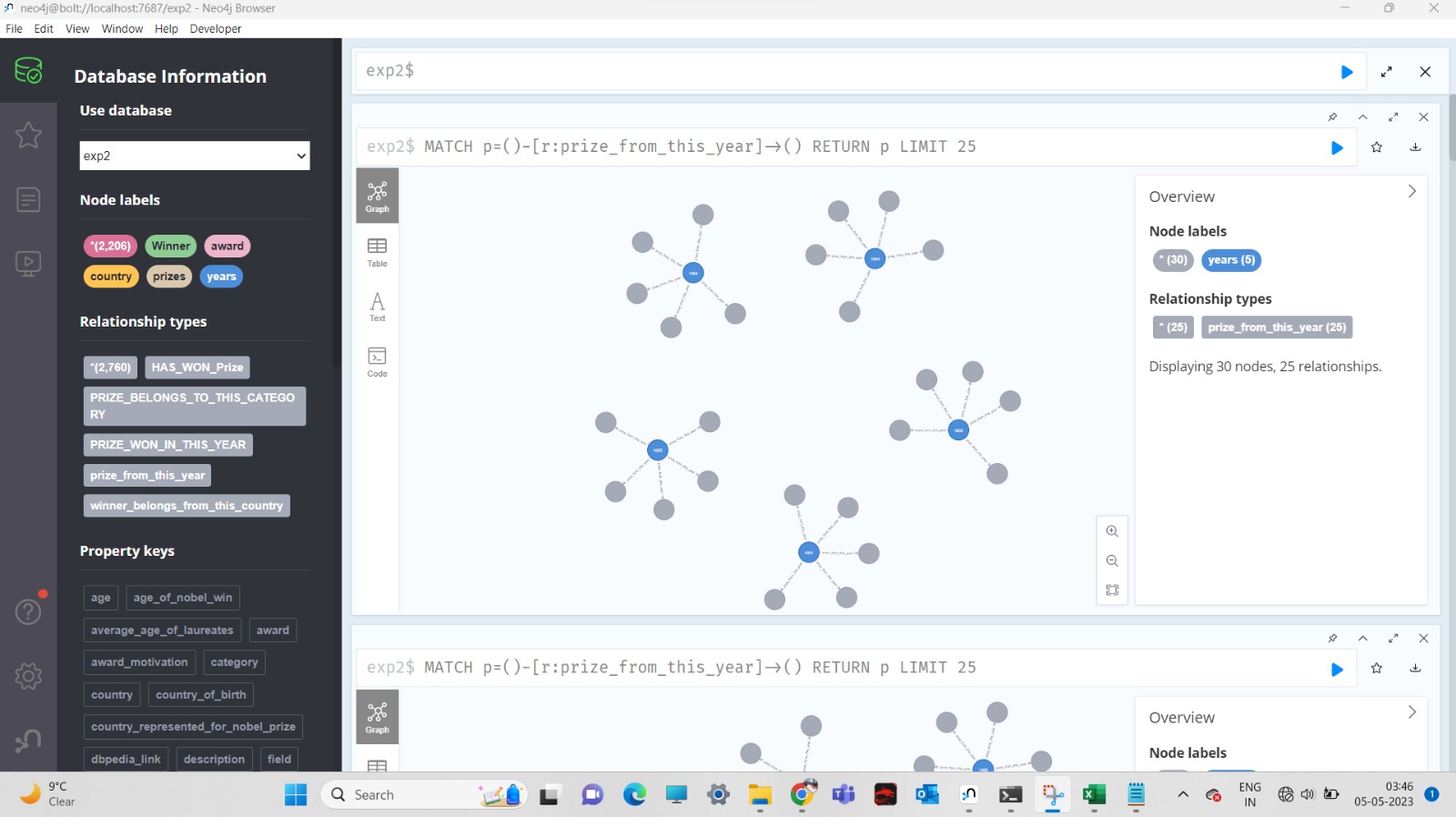


Output:

A screenshot of a computer

Description automatically generated with medium confidence

Deleting fake relationships created to demonstrate Delete Relationship operation.



OUTPUT:

A screenshot of a computer program

Description automatically generated with low confidence

Relationships deleted Successfully.

**Conclusion:**

Total 5 dataset which were in csv file format have been integrated in Graph neo4j database. Here we have created a database which is reliable and consistent. The relationships make it easy to relate the nodes present in the database. A GUI interface can be created for this database for END USER interaction by integrating the database with GUI interface.

This database can be used with various cloud platforms like AWS, Azure, GCP. If there was no budget constraint, we would I demonstrated on cloud platform like Azure where graph database performs really well. While running this database software we noticed that ever after having a RAM of 8gb and Graphics of 4gb in our computer. The software ran a bit slowly. In future I would like to use this database on cloud platform as cloud has more computing power.

As this was our first time using Graph database we learned a lot from this experience and would like to explore more and become proficient in this technology.