

Homework BDE 2

Raffaele Russo, Alessandro Vanacore, Giuliano Di Giuseppe

Indice

1	Traccia	2
1.1	MongoDB	2
1.1.1	Topic di ricerca	2
1.1.2	Subtopic di ricerca	3
1.2	Neo4J	4
1.2.1	Caricamento dataset e creazione nodi	5
1.2.2	Query: Topic di ricerca per anno	6
1.2.3	Query: Subtopic di ricerca per anno	6
1.2.4	Creazione nodi anno di inizio	7
1.2.5	Creazione relazioni tra progetti e anno di inizio	7
1.2.6	Query: Progetti iniziati in un dato intervallo temporale	8
1.2.7	Creazione nodi campo di ricerca e relazioni tra progetto e campo di ricerca	8
1.2.8	Query: Progetti afferenti a un dato campo di ricerca iniziati in un certo intervallo temporale	10
1.2.9	Creazione relazioni tra anno di inizio e campo di ricerca	10
1.2.10	Query: Campi di ricerca della Federico II in un certo anno	11
1.2.11	Creazione relazioni tra i nodi Topic e Subtopic	12
1.2.12	Creazione nodi ricercatori e doppia relazione tra progetti e ricercatori	12
1.2.13	Query: Progetti a cui ha lavorato un ricercatore	13
1.2.14	Query: Ricercatori che hanno lavorato a un progetto	14
1.2.15	Creazione relazione tra ciascun ricercatori e i propri ambiti di competenza	14
1.2.16	Query: Competenze di un ricercatore	14

1 Traccia

Importare i dati presenti sul dataset relativo ai progetti di ricerca della Federico II sia in MongoDB come collezione di documenti, sia in Neo4J come grafo. Eseguire poi una serie di analytics su entrambi i database nosql che descrivano nella maniera più efficace possibile le competenze della Federico II in termini di “topic” di ricerca affrontati nel corso degli anni.

1.1 MongoDB

Il codice seguente consente di importare il dataset relativo ai progetti di ricerca della Federico II in MongoDB come collezione di documenti JSON.

```
# Lettura CSV e inserimento documenti JSON
with open('dataset_header.csv', 'r') as file:
    csv_data = csv.DictReader(file, delimiter=",")
    for row in csv_data:
        # Inserisci ogni riga come documento JSON
        collection.insert_one(row)
```

Analizzando il campo *Fields of Research (ANZSRC 2020)* ci siamo resi conto che ciascun progetto potesse avere diverse competenze e che inoltre esse venissero descritte da un Topic generico, per esempio “Engineering”, e una sua specializzazione, “Control Engineering, Mechatronics and Robotics”. Per poterle distinguere, il Topic è descritto da un codice a doppia cifra, mentre un suo Subtopic presenta un codice a 4 cifre dove le prime sono quelle relative alla sua macroarea. L'utilizzo della pipeline consente di applicare sequenzialmente delle operazioni su di una collezione di dati con MongoDB.

1.1.1 Topic di ricerca

Il seguente codice comincia effettuando uno split sugli elementi del campo *Fields of Research (ANZSRC 2020)*, questo ci ha consentito di ottenere i vari topic di ricerca (Topic e Subtopic) per ogni riga del dataset. Dopo aver filtrato gli elementi nulli e vuoti sui campi *Fields of Research (ANZSRC 2020)* e *Start Year*, abbiamo cercato tutti gli elementi che cominciassero con sole due cifre, individuando così esclusivamente i Topic del progetto. A questo punto si è terminato effettuando un raggruppamento in base al Topic e in base all' Anno di Inizio del progetto.

```
pipeline = [
    {
        '$project': {
            'Fields of Research (ANZSRC 2020)': {
                '$split': ['$Fields of Research (ANZSRC 2020)', '']
            },
            'Start Year': 1
        }
    },
    {
        '$unwind': '$Fields of Research (ANZSRC 2020)'
    },
    {
        '$project': {
            'Fields of Research (ANZSRC 2020)': {
                '$trim': {
                    'input': '$Fields of Research (ANZSRC 2020)',
                    'chars': ' '
                }
            },
            'Start Year': 1
        }
    }
]
```

```

    },
    #filtriamo stringhe vuote e nulle
    {
        '$match': {
            'Fields of Research (ANZSRC 2020)': {'$ne': ''},
            'Start Year': {'$ne': ''}
        }
    },
    {
        '$match': {
            'Fields of Research (ANZSRC 2020)': {
                '$regex': '^d{2}\s'
            }
        }
    },
    ...
    #raggruppiamo sulla base di campo di ricerca e anno, calcoliamo il count
    {
        '$group': {
            '_id': {
                'Topic': '$Fields of Research (ANZSRC 2020)',
                'Year': '$Start Year'
            },
            'Count': {'$sum': 1}
        }
    },
    #ordiniamo in base all'anno di inizio
    {
        '$sort': {'_id.Year': 1}
    }
]
result = collection.aggregate(pipeline)

```

I risultati ottenuti sono stati ordinati in maniera crescente in base allo Start Year.

```

Topic: 49 Mathematical Sciences, year: 1980, count: 1
Topic: 51 Physical Sciences, year: 1980, count: 1
Topic: 40 Engineering, year: 1980, count: 1
Topic: 30 Agricultural, Veterinary and Food Sciences, year: 1985, count: 1
Topic: 31 Biological Sciences, year: 1986, count: 8
Topic: 30 Agricultural, Veterinary and Food Sciences, year: 1986, count: 1
Topic: 40 Engineering, year: 1986, count: 1
Topic: 37 Earth Sciences, year: 1987, count: 5
Topic: 31 Biological Sciences, year: 1987, count: 1
Topic: 40 Engineering, year: 1988, count: 1
Topic: 48 Law and Legal Studies, year: 1988, count: 1
Topic: 44 Human Society, year: 1988, count: 1
Topic: 31 Biological Sciences, year: 1988, count: 1
Topic: 37 Earth Sciences, year: 1988, count: 2
Topic: 34 Chemical Sciences, year: 1988, count: 1
Topic: 47 Language, Communication and Culture, year: 1988, count: 1
Topic: 51 Physical Sciences, year: 1989, count: 1
Topic: 49 Mathematical Sciences, year: 1989, count: 1
Topic: 46 Information and Computing Sciences, year: 1989, count: 1
Topic: 40 Engineering, year: 1989, count: 1
Topic: 46 Information and Computing Sciences, year: 1990, count: 1
Topic: 38 Economics, year: 1990, count: 1
Topic: 31 Biological Sciences, year: 1990, count: 1
Topic: 40 Engineering, year: 1990, count: 5
Topic: 33 Built Environment and Design, year: 1990, count: 1
...
Topic: 40 Engineering, year: 2023, count: 5
Topic: 41 Environmental Sciences, year: 2023, count: 5
Topic: 42 Health Sciences, year: 2023, count: 1
Topic: 48 Law and Legal Studies, year: 2023, count: 1

```

1.1.2 Subtopic di ricerca

```

pipeline = [
    {
        '$project': {
            'Fields of Research (ANZSRC 2020)': {
                '$split': ['$Fields of Research (ANZSRC 2020)', ';']
            },
            'Start Year': 1
        }
    },
    {
        '$unwind': '$Fields of Research (ANZSRC 2020)'
    },
    {

```

```

        '$project': {
            'Fields of Research (ANZSRC 2020)': {
                '$trim': {
                    'input': '$Fields of Research (ANZSRC 2020)',
                    'chars': ' '
                }
            },
            'Start Year': 1
        }
    },
    #filtriamo stringhe vuote e nulle
    {
        '$match': {
            'Fields of Research (ANZSRC 2020)': {'$ne': ''},
            'Start Year': {'$ne': ''}
        }
    },
    {
        '$match': {
            'Fields of Research (ANZSRC 2020)': {
                '$regex': '^\\d{4}\\s'
            }
        }
    },
    #raggruppiamo sulla base di campo di ricerca e anno, calcoliamo il count
    {
        '$group': {
            '_id': {
                'Subtopic': '$Fields of Research (ANZSRC 2020)',
                'Year': '$Start Year'
            },
            'Count': {'$sum': 1}
        }
    },
    #ordiniamo in base all'anno di inizio
    {
        '$sort': {'_id.Year': 1}
    }
]
result = collection.aggregate(pipeline)

```

I risultati ottenuti sono stati ordinati in maniera crescente in base allo Start Year.

```

Subtopic: 5106 Nuclear and Plasma Physics, year: 1980, count: 1
Subtopic: 4902 Mathematical Physics, year: 1980, count: 1
Subtopic: 3008 Horticultural Production, year: 1985, count: 1
Subtopic: 3004 Crop and Pasture Production, year: 1985, count: 1
Subtopic: 3108 Plant Biology, year: 1986, count: 1
Subtopic: 4002 Automotive Engineering, year: 1986, count: 1
Subtopic: 3101 Biochemistry and Cell Biology, year: 1986, count: 3
Subtopic: 4017 Mechanical Engineering, year: 1986, count: 1
Subtopic: 3106 Industrial Biotechnology, year: 1986, count: 4
Subtopic: 3708 Oceanography, year: 1987, count: 1
Subtopic: 3101 Biochemistry and Cell Biology, year: 1987, count: 1
Subtopic: 3705 Geology, year: 1987, count: 2
Subtopic: 3709 Physical Geography and Environmental Geoscience, year: 1987, count: 4
Subtopic: 3101 Biochemistry and Cell Biology, year: 1988, count: 1
Subtopic: 3705 Geology, year: 1988, count: 2
Subtopic: 4016 Materials Engineering, year: 1988, count: 1
Subtopic: 4407 Policy and Administration, year: 1988, count: 1
Subtopic: 4702 Cultural Studies, year: 1988, count: 1
Subtopic: 3709 Physical Geography and Environmental Geoscience, year: 1988, count: 2
Subtopic: 3403 Macromolecular and Materials Chemistry, year: 1988, count: 1
Subtopic: 4012 Fluid Mechanics and Thermal Engineering, year: 1989, count: 1
Subtopic: 4904 Pure Mathematics, year: 1989, count: 1
Subtopic: 5106 Nuclear and Plasma Physics, year: 1989, count: 1
Subtopic: 3106 Industrial Biotechnology, year: 1990, count: 1
Subtopic: 4004 Chemical Engineering, year: 1990, count: 1
...
Subtopic: 4410 Sociology, year: 2023, count: 1
Subtopic: 4803 International and Comparative Law, year: 2023, count: 1
Subtopic: 4904 Pure Mathematics, year: 2023, count: 1
Subtopic: 3705 Geology, year: 2023, count: 1

```

1.2 Neo4J

Il seguente codice ha permesso di importare il dataset di nostro interesse e di generare un nodo per ogni row di quest'ultimo dove ogni colonna del dataset di partenza è una sua property, come visibile in figura 1.

1.2.1 Caricamento dataset e creazione nodi

```
LOAD CSV WITH HEADERS FROM 'file:///dataset_header.csv' AS row FIELDTERMINATOR '|'
CREATE (g:Grant {
  Rank: toInteger(row.Rank),
  GrantID: row.`Grant ID`,
  GrantNumber: row.`Grant Number`,
  Title: row.Title,
  TitleTranslated: row.`Title translated`,
  Abstract: row.Abstract,
  AbstractTranslated: row.`Abstract translated`,
  Keywords: row.Keywords,
  FundingAmount: toFloat(row.`Funding Amount`),
  Currency: row.Currency,
  FundingAmountInEUR: toFloat(row.`Funding Amount in EUR`),
  StartDate: row.`Start Date`,
  StartYear: toInteger(row.`Start Year`),
  EndDate: toInteger(row.`End Date`),
  EndYear: toInteger(row.`End Year`),
  Researchers: row.Researchers,
  ResearchOrganizationOriginal: row.`Research Organization original`,
  ResearchOrganizationStandardized: row.`Research Organization standardized`,
  GRIDID: row.`GRID ID`,
  CityOfResearchOrganization: row.`City of Research organization`,
  StateOfResearchOrganization: row.`State of Research organization`,
  CountryOfResearchOrganization: row.`Country of Research organization`,
  Funder: row.Funder,
  FunderGroup: row.`Funder Group`,
  FunderCountry: row.`Funder Country`,
  Program: row.Program,
  ResultingPublications: row.`Resulting publications`,
  SourceLinkout: row.`Source Linkout`,
  DimensionsURL: row.`Dimensions URL`,
  FieldsOfResearchANZSRC2020: row.`Fields of Research (ANZSRC 2020)`,
  RCDG_Categories: row.`RCDG Categories`,
  HRCS_HC_Categories: row.`HRCS HC Categories`,
  HRCS_RAC_Categories: row.`HRCS RAC Categories`,
  CancerTypes: row.`Cancer Types`,
  CSO_Categories: row.`CSO Categories`,
  UnitsOfAssessment: row.`Units of Assessment`,
  SustainableDevelopmentGoals: row.`Sustainable Development Goals`
})
```

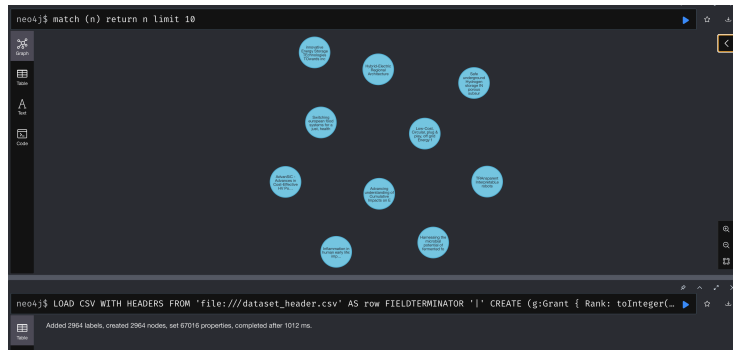
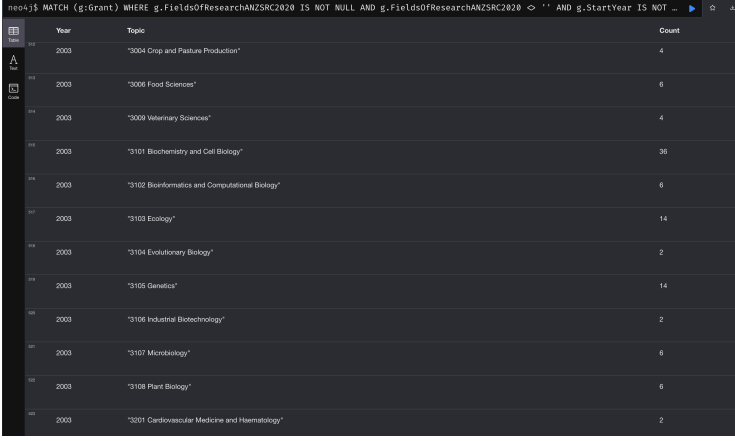


Figura 1: Overview Grafo

1.2.2 Query: Topic di ricerca per anno

```
#Ricerca topic per anno
MATCH (g:Grant)
WHERE g.FieldsOfResearchANZSRC2020 IS NOT NULL
AND g.FieldsOfResearchANZSRC2020 <> ''
AND g.StartYear IS NOT NULL
AND g.StartYear <> ''
WITH g.StartYear AS StartYear, SPLIT(g.FieldsOfResearchANZSRC2020, ';') AS fields
UNWIND fields AS field
WITH StartYear AS Year, TRIM(field) AS Topic
WHERE Topic =~ '^\\d{2} .+'
RETURN Year, Topic, COUNT(*) AS Count
ORDER BY Year, Topic
```



Year	Topic	Count
2003	"3004 Crop and Pasture Production"	4
2003	"3008 Food Sciences"	6
2003	"3009 Veterinary Sciences"	4
2003	"3101 Biochemistry and Cell Biology"	38
2003	"3102 Bioinformatics and Computational Biology"	6
2003	"3103 Ecology"	14
2003	"3104 Evolutionary Biology"	2
2003	"3105 Genetics"	14
2003	"3106 Industrial Biotechnology"	2
2003	"3107 Microbiology"	6
2003	"3108 Plant Biology"	6
2003	"3201 Cardiovascular Medicine and Hematology"	2

Figura 2: Risultati Topic

1.2.3 Query: Subtopic di ricerca per anno

```
#Ricerca subtopic per anno
MATCH (g:Grant)
WHERE g.FieldsOfResearchANZSRC2020 IS NOT NULL
AND g.FieldsOfResearchANZSRC2020 <> ''
AND g.StartYear IS NOT NULL
AND g.StartYear <> ''
WITH g.StartYear AS StartYear, SPLIT(g.FieldsOfResearchANZSRC2020, ';') AS fields
UNWIND fields AS field
WITH StartYear AS Year, TRIM(field) AS Subtopic
WHERE Subtopic =~ '^\\d{4} .+'
RETURN Year, Subtopic, COUNT(*) AS Count
ORDER BY Year, Subtopic
```

```
neokj$ MATCH (g:Grant) WHERE g.FieldsOfResearchANZSRC2020 IS NOT NULL AND g.FieldsOfResearchANZSRC2020 <> '' AND g.StartYear IS NOT NULL
```

	Year	Topic	Count
100	2003	"3004 Crop and Pasture Production"	4
100	2003	"3006 Food Sciences"	6
100	2003	"3009 Veterinary Sciences"	4
100	2003	"3101 Biochemistry and Cell Biology"	36
100	2003	"3102 Bioinformatics and Computational Biology"	6
100	2003	"3103 Ecology"	14
100	2003	"3104 Evolutionary Biology"	2
100	2003	"3105 Genetics"	14
100	2003	"3106 Industrial Biotechnology"	2
100	2003	"3107 Microbiology"	6
100	2003	"3108 Plant Biology"	6
100	2003	"3201 Cardiovascular Medicine and Hematology"	2

Figura 3: Risultati Subtopic

1.2.4 Creazione nodi anno di inizio

```
MATCH (g:Grant)
WITH DISTINCT g.StartYear AS startYear
WHERE startYear IS NOT NULL
MERGE (y:StartYear {year: startYear})
```

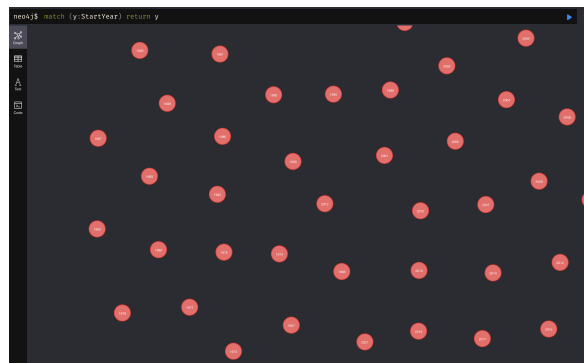


Figura 4: Nodi anno di inizio

1.2.5 Creazione relazioni tra progetti e anno di inizio

```
MATCH (g:Grant), (y:StartYear {year: g.StartYear})
CREATE (g)-[:STARTS_IN]->(y)
```

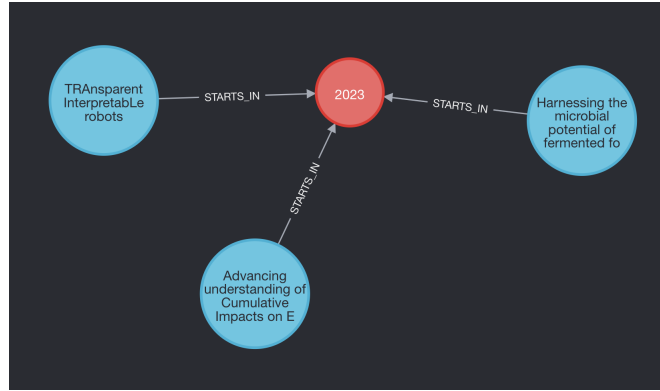


Figura 5: Relazioni progetti e anno di inizio

1.2.6 Query: Progetti iniziati in un dato intervallo temporale

```
MATCH (g:Grant) -[:STARTS_IN]->(y:StartYear)
WHERE y.year >= 1980 AND y.year <= 1985
RETURN g, y
```



Figura 6: Progetti iniziati in un dato intervallo temporale

1.2.7 Creazione nodi campo di ricerca e relazioni tra progetto e campo di ricerca

```
MATCH (g:Grant)
WITH g, split(g.FieldsOfResearchANZSRC2020, ';') AS fields
UNWIND fields AS field
WITH DISTINCT g, TRIM(field) AS trimmedField
WHERE trimmedField IS NOT NULL AND trimmedField <> ''
MERGE (f:FieldOfResearch {name: trimmedField})
MERGE (g) -[:HAS_FIELD]->(f)
```

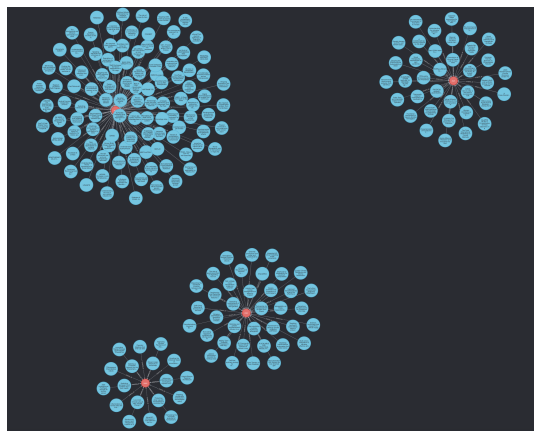



Figura 7: Progetti iniziati in un dato intervallo temporale

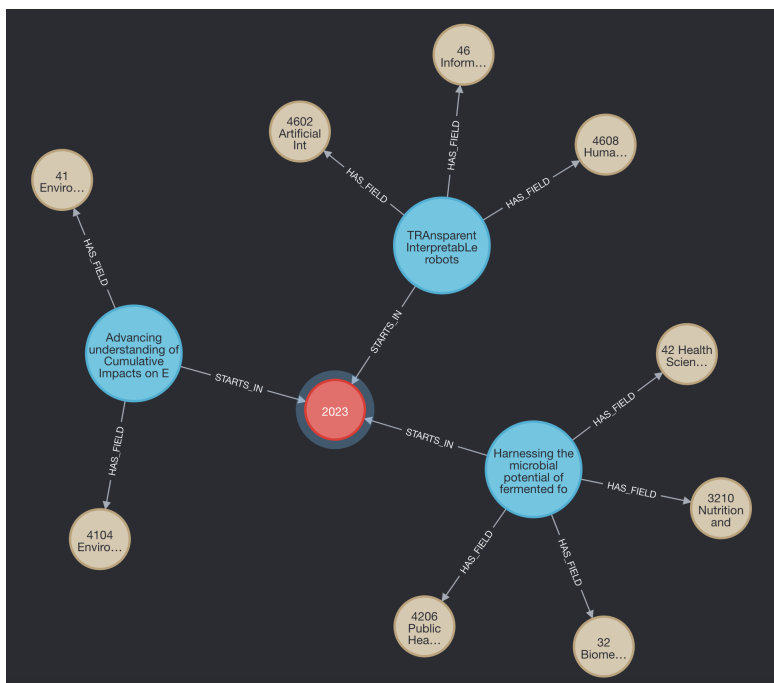


Figura 8: Relazioni tra progetti e campi di ricerca

1.2.8 Query: Progetti afferenti a un dato campo di ricerca iniziati in un certo intervallo temporale

```
MATCH (g:Grant)-[:HAS_FIELD]->(f:FieldOfResearch)
WHERE f.name = '40 Engineering' AND g.StartYear >= 1990 AND g.StartYear <= 1995
RETURN g, f
```

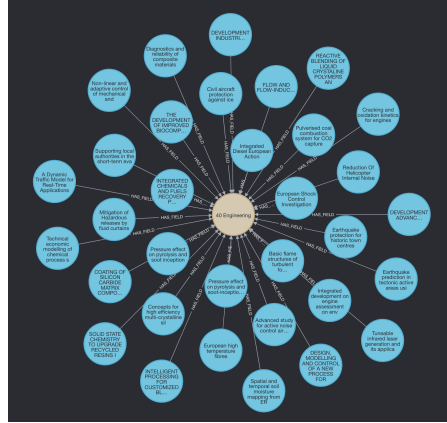


Figura 9: Progetti afferenti a un dato campo di ricerca iniziati in un certo intervallo temporale

1.2.9 Creazione relazioni tra anno di inizio e campo di ricerca

```
MATCH (g:Grant)-[:STARTS_IN]->(y:StartYear),
(g:Grant)-[:HAS_FIELD]->(f:FieldOfResearch)
MERGE (y)-[:GRANT]->(f)
```

1.2.10 Query: Campi di ricerca della Federico II in un certo anno

```
MATCH (g:Grant)-[:HAS_FIELD]->(f:FieldOfResearch)
WHERE g.StartYear = 1980
MATCH (g)-[:STARTS_IN]->(y:StartYear)
RETURN DISTINCT f, y
```

Possiamo utilizzare questa nuova query per valutare come variano gli ambiti di studio trattati dalla Federico II in due anni differenti.

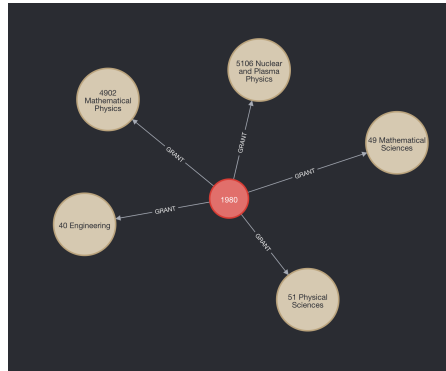


Figura 10: Ambito di studio trattati nel 1980



Figura 11: Ambiti di studio trattati nel 2020

1.2.11 Creazione relazioni tra i nodi Topic e Subtopic

```
MATCH (f1:FieldOfResearch)
WHERE f1.name =~ '^[0-9]{2} .*'
WITH f1
MATCH (f2:FieldOfResearch)
WHERE f2.name =~ (LEFT(f1.name, 2) + '.*') AND f2 <> f1
CREATE (f1)-[:SPECIALIZE]->(f2)
```

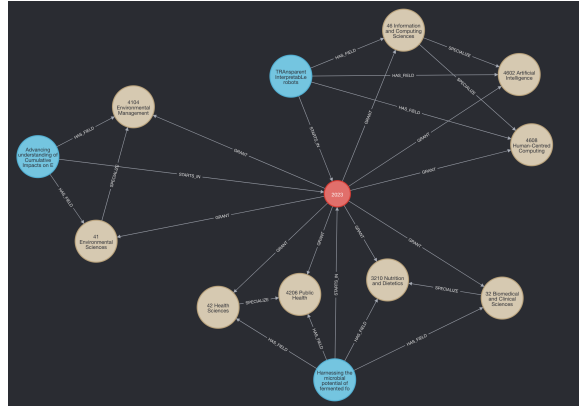


Figura 12: Relazioni Topic e Subtopic

1.2.12 Creazione nodi ricercatori e doppia relazione tra progetti e ricercatori

```
MATCH (g:Grant)
WITH g, split(g.Researchers, ',') AS researchers
UNWIND researchers AS researcher
WITH DISTINCT g, TRIM(researcher) AS trimmedResearcher
WHERE trimmedResearcher IS NOT NULL AND trimmedResearcher <> ''
MERGE (r:Researcher {name: trimmedResearcher})
MERGE (r)-[:HAS_WORKED]->(g)
MERGE (g)-[:RESEARCHER]->(r)
```

1.2.13 Query: Progetti a cui ha lavorato un ricercatore

```
MATCH (r:Researcher {name: 'BARLETTA Antonio'}) -[:HAS_WORKED]->(g:Grant)  
RETURN r, g
```



Figura 13: Progetti di un ricercatore

1.2.14 Query: Ricercatori cha hanno lavorato a un progetto

```
MATCH (g:Grant {GrantID: 'grant.13018646'})-[:RESEARCHER]->(r:Researcher)
RETURN g, r
```

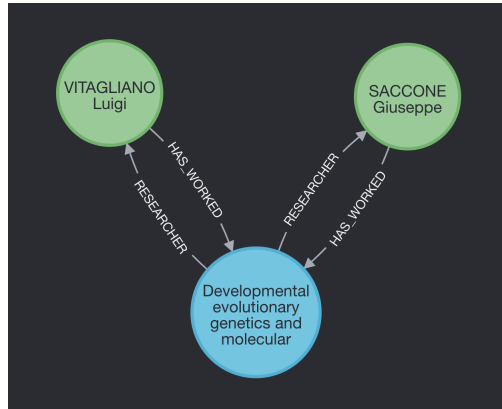


Figura 14: Ricercatori di un progetto

1.2.15 Creazione relazione tra ciascun ricercatori e i propri ambiti di competenza

```
MATCH (g:Grant)-[:HAS_FIELD]->(f:FieldOfResearch)
WITH g, split(g.Researchers, ',') AS researchers, collect(f) AS fields
UNWIND researchers AS researcher
WITH DISTINCT g, TRIM(researcher) AS trimmedResearcher, fields
WHERE trimmedResearcher IS NOT NULL AND trimmedResearcher <> ''
MERGE (r:Researcher {name: trimmedResearcher})
FOREACH (field IN fields | MERGE (r)-[:WORKS_ON]->(field))
```

1.2.16 Query: Competenze di un ricercatore

```
MATCH (r:Researcher {name: 'SILINGARDI Vittorio'})-[:WORKS_ON]->(f:FieldOfResearch)
RETURN r, f
```



Figura 15: Competenze di un ricercatore