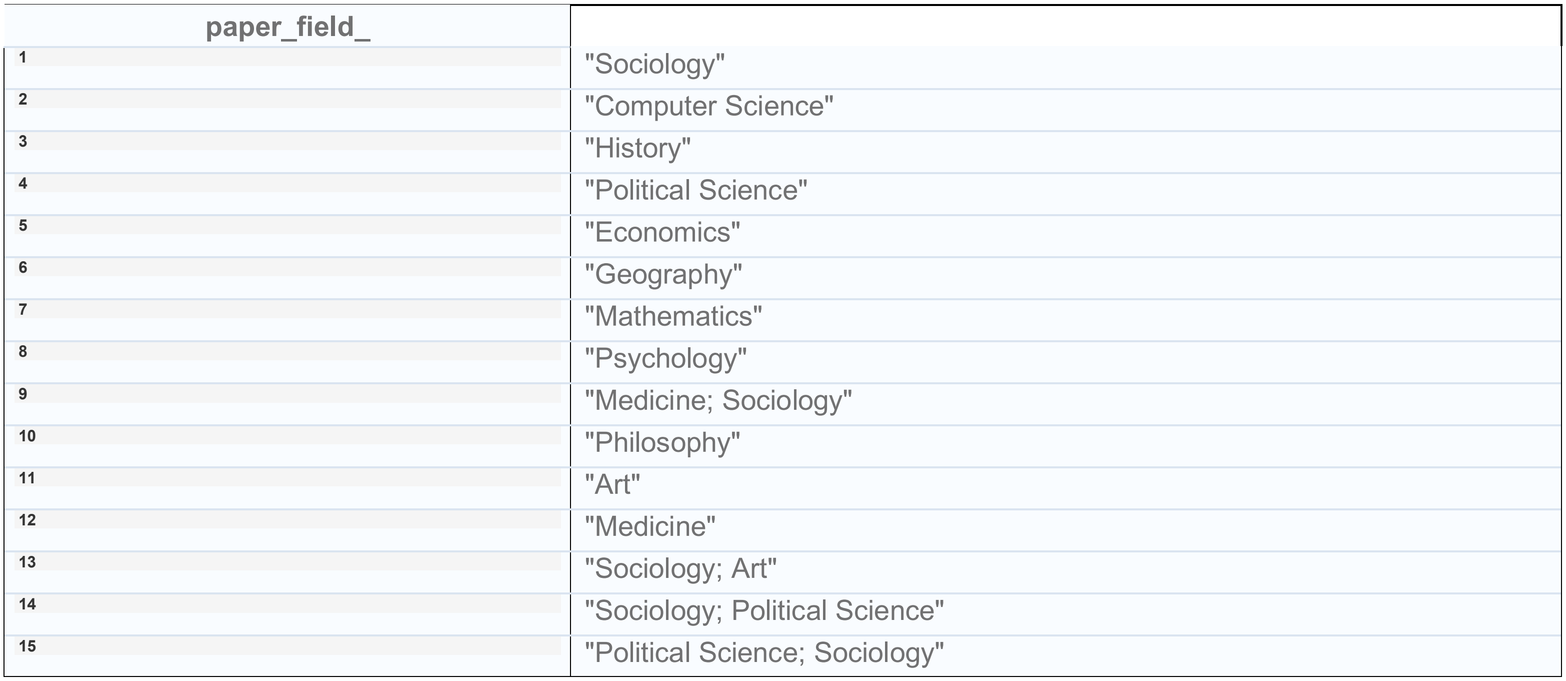
GDS_Project_Queries

```
// Unique constraint for Author_id

CREATE CONSTRAINT IF NOT EXISTS FOR (a:Author) REQUIRE a.Author_id IS UNIQUE;


// Unique constraint for Paper_id

CREATE CONSTRAINT IF NOT EXISTS FOR (p:Paper) REQUIRE p.Paper_id IS UNIQUE;


CALL apoc.periodic.iterate(
 "LOAD CSV WITH HEADERS FROM 'file:///cleaned_author_data.csv'
AS row RETURN row",
 "MERGE (a:Author{Author_id: row.`Author ID`, Author_name: row.`Author Name`,
Author_Url: row.`Author URL`})",
 {batchSize: 2000}
);



CALL apoc.periodic.iterate(
  "LOAD CSV WITH HEADERS FROM 'file:///cleaned_paper_data_valid_rows_.csv' AS row
   RETURN row",
  "MERGE (p:Paper {Paper_id: row.`Paper ID`,
              Paper_doi: row.`Paper DOI`,
              Paper_url: row.`Paper URL`,
              Paper_title: row.`Paper Title`,
              Paper_year: toInteger(row.`Paper Year`),
              Paper_citation_count: toInteger(row.`Paper Citation Count`),
              Paper_field: row.`Fields of Study`,
              Paper_journal_volume: toFloat(row.`Journal Volume`),
              Paper_journal_date: row.`Journal Date`})",
  {batchSize: 2000}
);
```

```
CALL apoc.periodic.iterate( "LOAD CSV WITH HEADERS FROM
'file:///cleaned_journal_data.csv'

AS row RETURN row",

 "MERGE (j:Journal {Journal_Name: row.`Journal Name`, Journal_Publisher: row.`Journal
Publisher`})",

 {batchSize: 2000}

);



CALL apoc.periodic.iterate(
 "LOAD CSV WITH HEADERS FROM 'file:///cleaned_topic.csv'
AS row RETURN row",
 "MERGE (t:Topic{Topic_ID: row.`Topic ID`, Topic_Name: row.`Topic Name`,
Topic_URL:row.`Topic URL`})",
 {batchSize: 2000}
);


CALL apoc.periodic.iterate(
 "LOAD CSV WITH HEADERS FROM
'file:///cleaned_author_paper.csv' AS row RETURN row",
 "MATCH (a:Author {Author_id: row.`Author ID`}) MATCH (p:Paper {Paper_id: row.`Paper
ID`})
 MERGE  (a)-[:AUTHORED]->(p)",
 {batchSize: 2000}
);


CALL apoc.periodic.iterate(
  "LOAD CSV WITH HEADERS FROM 'file:///cleaned_paper_reference.csv' AS row RETURN
row",
  "MATCH (p:Paper {Paper_id: row.`Paper ID`})
   MATCH (p1:Paper {Paper_id: row.`Referenced Paper ID`})
  MERGE (p)-[:CITES]->(p1)",
 {batchSize: 2000}
);

CALL apoc.periodic.iterate(
 "LOAD CSV WITH HEADERS FROM 'file:///cleaned_paper_topic.csv' AS row RETURN row",
 "MATCH (p:Paper {Paper_id: row.`Paper ID`})
  MATCH (t:Topic {Topic_id: row.`Topic ID`})
  MERGE (p)-[:HAS_TOPIC]->(t)",
 {batchSize: 2000}
);


CALL apoc.periodic.iterate(
 "LOAD CSV WITH HEADERS FROM 'file:///cleaned_paper_journal.csv'
AS row RETURN row",
 "MATCH (p:Paper {Paper_id: row.`Paper ID`})
  MATCH (j:Journal {Journal_Name: row.`Journal Name`})
  MERGE (p)-[:PUBLISHED_IN]->(j)",
 {batchSize: 2000}
);
```

```
Match(p:Paper) return distinct p.Paper_field as paper_field_;
```

Result:



paper_field_	
1	"Sociology"
2	"Computer Science"
3	"History"
4	"Political Science"
5	"Economics"
6	"Geography"
7	"Mathematics"
8	"Psychology"
9	"Medicine; Sociology"
10	"Philosophy"
11	"Art"
12	"Medicine"
13	"Sociology; Art"
14	"Sociology; Political Science"
15	"Political Science; Sociology"

16	"Sociology; History"
17	"Psychology; Sociology"
18	"Geography; Economics"
19	"Sociology; Medicine"
20	"Economics; Medicine"
21	"Biology; Sociology"
22	"Psychology; Geography"
23	"Economics; Political Science"
24	"Sociology; Geography"
25	"Economics; Psychology"
26	"Psychology; Political Science"
27	"Philosophy; Medicine"
28	"Sociology; Geography; Medicine"
29	"Economics; Medicine; Engineering"
30	"Economics; Medicine; Geography"
31	"Political Science; Medicine"
32	"Sociology; Economics"
33	"Political Science; Geography; Medicine"
34	"Economics; Medicine; Political Science"
35	"Political Science; Psychology"
36	"Geography; Medicine"
37	"Art; Medicine; Political Science"
38	"Sociology; Psychology"
39	"Economics; Medicine; Sociology"
40	"Biology"
41	"Geography; History; Sociology"
42	"Geography; Political Science"
43	"Political Science; Art"
44	"Political Science; Sociology; Medicine"
45	"Economics; Geography"
46	"Sociology; Medicine; Political Science"
47	"Psychology; Medicine"
48	"History; Sociology"
49	"Geography; Medicine; Political Science"
50	"Political Science; Economics"
51	"Economics; Mathematics"
52	"Sociology; Political Science; Medicine"
53	"Sociology; Biology"
54	"Economics; Sociology"
55	"Philosophy; Sociology"
56	"Medicine; Economics"
57	"History; Medicine"
58	"Geography; Political Science; Medicine"
59	"Art; Medicine"
60	"Art; Geography; Medicine"

61	"Geography; Sociology"
62	"Geography; Medicine; Sociology"
63	"Mathematics; Psychology"
64	"Psychology; Economics"
65	"Philosophy; Art"
66	"Geography; Art; Medicine"
67	"Medicine; Psychology"
68	"Medicine; Geography"
69	"Medicine; Computer Science"
70	"Political Science; Art; Medicine"
71	"History; Economics"
72	"Sociology; Business"
73	"Business"
74	"Computer Science; Sociology"
75	"Psychology; Medicine; Sociology"
76	"Political Science; Computer Science; Sociology; Psychology"
77	"Geography; Materials Science"
78	"Sociology; Business; Political Science"
79	"Sociology; Mathematics"
80	"Business; Sociology"
81	"Economics; Computer Science"
82	"Economics; Business"
83	"Computer Science; Psychology"
84	"Sociology; Economics; Geography"
85	"Biology; Economics"
86	"Physics"
87	"Medicine; Psychology; Political Science"
88	"Political Science; Business"
89	"Sociology; Physics"
90	"Engineering; Sociology"
91	"Sociology; Engineering"
92	"Political Science; Geography"
93	"Engineering"
94	"Geography; Biology"
95	"Medicine; Sociology; Geography"
96	"Chemistry; Sociology"
97	"Political Science; Medicine; Psychology"
98	"Mathematics; Sociology"
99	"Geography; Sociology; Political Science"
100	"Geography; Chemistry"
101	"Business; Medicine"
102	"Geography; Sociology; Economics"
103	"Engineering; History"
104	"Economics; Political Science; Medicine"
105	"Psychology; Engineering; Geography"

106	"History; Geography"
107	"Psychology; Business"
108	"Sociology; Political Science; Economics"
109	"Computer Science; Medicine"
110	"Business; Economics"
111	"Geology"
112	"Environmental Science; Medicine"
113	"Biology; Medicine"
114	"Medicine; Business"
115	"Sociology; Medicine; Biology"
116	"Psychology; Chemistry"
117	"Geography; Economics; Business"
118	"Geography; Engineering"
119	"Environmental Science"
120	"Art; Sociology"
121	"Environmental Science; Medicine; Geography"
122	"Materials Science"
123	"Business; Engineering"
124	"Psychology; Medicine; Political Science"
125	"Medicine; Psychology; Art"
126	"Geography; Psychology"
127	"Political Science; Psychology; Medicine"
128	"Geography; Medicine; Economics"
129	"Medicine; Political Science"
130	"Business; Political Science"
131	"Political Science; Medicine; Economics"
132	"Mathematics; Physics"
133	"Psychology; History"

Social Sciences:

- **Sociology**
- **Political Science**
- **Economics**
- **Psychology**
- **History**
- **Business**

Natural Sciences:

- **Geography** (natural sciences context, such as physical geography)
- **Mathematics**
- **Physics**
- **Chemistry**
- **Geology**

Life Sciences & Medicine:

- **Medicine**
- **Biology**
- **Environmental Science**

Engineering & Technology:

- **Engineering**
- **Computer Science**
- **Materials Science**

Arts & Humanities:

- **Art**
- **Philosophy**

```
// Step 2: Relate journals to papers, collect paper fields, map to categories, ensure uniqueness, and set properties
```

```
MATCH (j:Journal)<-[:PUBLISHED_IN]-(p:Paper)
WHERE p.Paper_field IS NOT NULL
WITH j, collect(DISTINCT p.Paper_field) AS fields
WITH j, [field IN fields | split(field, ';')] AS field_lists
WITH j, [field IN apoc.coll.flatten(field_lists) | trim(field)] AS all_paper_fields
WITH j, all_paper_fields,
  [cat IN ['Social Sciences', 'Natural Sciences', 'Life Sciences & Medicine',
'Engineering & Technology', 'Arts & Humanities']
   WHERE ANY(field IN all_paper_fields WHERE
      (cat = 'Social Sciences' AND field IN ['Sociology', 'Political Science',
'Economics', 'Psychology', 'History', 'Business']) OR
      (cat = 'Natural Sciences' AND field IN ['Geography', 'Mathematics', 'Physics',
'Chemistry', 'Geology']) OR
      (cat = 'Life Sciences & Medicine' AND field IN ['Medicine', 'Biology',
'Environmental Science']) OR
      (cat = 'Engineering & Technology' AND field IN ['Engineering', 'Computer
Science', 'Materials Science']) OR
      (cat = 'Arts & Humanities' AND field IN ['Art', 'Philosophy']))] AS
applicable_categories
WITH j, all_paper_fields, apoc.coll.toSet(applicable_categories) AS
unique_categories
SET j.categories = all_paper_fields
SET j.category = CASE
  WHEN size(unique_categories) = 0 THEN 'Interdisciplinary'
  WHEN size(unique_categories) = 1 THEN unique_categories[0]
  ELSE reduce(s = '', cat IN unique_categories | s + cat + '/')
END;

// Step 3: Fix trailing '/'
MATCH (j:Journal)
WHERE j.category IS NOT NULL AND toString(j.category) ENDS WITH '/'
SET j.category = left(toString(j.category), size(toString(j.category)) - 1);
```