

PublicationDatabaseLCN Project Direction Overview

For my project in this course I intend to develop personal websites for three of the Principal Investigators (PIs) in my lab - Bruce Fischl, André van der Kouwe and Doug Greve. Each PI's website will contain a record of the journal articles they have each published over the course of their research careers. A personal website is a very valuable tool for a scientist, as it is a marketing tool for their life's work. A website that is well organized, gives viewers an overview of your research interests, current projects and published articles and can lead to a wealth of opportunities. Oftentimes a scientist's website will determine whether potential doctoral graduate students, postdoctoral candidates and collaborators will reach out to the scientist expressing interest in their research and their excitement to work with them. It can be difficult to find all the aspects outlining a scientist's research career in one place, which is why I think personal websites for Bruce, André and Doug will be incredibly beneficial.

The information that will be housed on the website, specifically related to the PI's publications, will be the authors, publication name, publication year, journal, number of citations, DOI and link to publication. Typically on a researcher's website publications are listed according to date, with the most recent on the top. Official APA citation is typically used, as follows:

Dale, A.M., Fischl, B., Sereno, M.I. (1999).Cortical surface-based analysis: I. Segmentation and surface reconstruction. *NeuroImage*,9(2) 179-194.

Given the fact that Bruce alone currently has 305 publications (though many of which overlap with André and Doug), a long list of publication text citations will most likely not be very meaningful to website viewers. My goal is to make the websites interactive and not just simply a list of all the articles they have published over the years. Persons visiting a PIs website will be able to sort the publications by first author last name, publication year and number of citations the publication has amassed. Both the publication year and the citations will be sortable in both ascending and descending order. Users will also be able to use the search bar to look for specific publications by name, or publications from specific journals. The website will also be automatically updated with new publications and current citation values.

I recognize that a significant amount of programming is required in order to develop the website and ensure that publications are automatically updated. However my focus for this course is to develop a database that will house publication data (author's first and last names, publication name, publication date, citations etc) for Bruce, André and Doug, without redundancies. The database will need to identify which papers are relevant for the specific PI and provide that information to the web application. The database will also need to provide other queries such as sorting and text identification. I'm hoping I can gain the skills during this course to make this database, as well as get any feedback on different strategies I should implement.

Use Cases, Relevant Fields and Structural Rules

Use case 1

- A program searches for new papers published by Bruce, André and Doug, then adds the new publication information to the database. Other relevant fields e.g. citations should be updated.

Update Database with new information

- 1) A program is developed in python that automatically searches Pubmed for new publications.
- 2) The program obtains the APA reverence and adds the relevant information to the database.
- 3) All citation values in the database are compared with current citations on Pubmed and incorrect values are updated.

Step 1 is not directly relevant to the database. Though steps 2 and 3 involve directly interacting with the database, much of the work will be done by the python program. In these steps data is simply being compared and updated in the database. There is no real need to focus on the entities and how they relate to each other in this step. If the journal's impact factor has changed this will also be updated in the database. While the entity **Paper** can be identified in this use case it is not directly relevant in this use case as the main operations are done by the python program.

For now Full_Reference will be considered a field in Paper instead of a separate entity. This is because it is a one to one relationship with Paper with no optionality for multiple participations because a paper can only have one full reference (singular). If the database were to house full references in different styles ie APA, MLA etc, then this would change the relationship to a paper that can have one or many full references (plural). In this case Full_reference would need to be a separate entity. However there is no current plan for this database to house several different referencing methods.

| Field | What it Stores | Why it's Needed |
|------------|---|--|
| First_name | This is the first name of the author/s of the research paper. | This is needed to effectively identify the author, and to create an official APA citation. |
| Last_name | This is the last name of the author. | This is needed to effectively identify the author and to enable articles to be sorted by first author last name. |

| | | |
|----------------|---|--|
| Paper_year | This is the year the article was published. | This is needed to help identify the journal article, sort the articles by year and create an APA citation. |
| Paper_name | This is the name of the publication. | This is needed to identify the article, create the APA citation and allow the article to be identified in the search bar of the website. |
| Journal_name | This is the journal in which the article was published | This is needed to identify the article, create the APA citation and allow articles from specific journals to be searched on the website. |
| Full_reference | This is the full APA reference for the paper. | This is needed so that the end user can cite the paper in other papers, grants and conference submissions. |
| Impact_factor | This is the impact factor of the journal. | This tells the end user how influential the journal is. |
| Citation_num | This is the number of times the article has been cited in other journal articles. | This is needed to determine the articles impact and allow articles to be sorted by citation. |
| DOI | This is the article's publication ID that makes it identifiable on the internet. | This is an effective way to identify each article, both in the database and online. |
| Paper_link | This is the direct link to the journal article. | This will also be useful in allowing website users to go find the publication online. |
| PMCID | This is a code used by the NIH to identify papers. | This is needed if an end user is using the paper as a reference in grants or for conferences. |
| PMID | This is a code used by the NIH to identify papers. | This is needed if an end user is using the paper as a reference in grants or for conferences. |

Use case 2

- Database is accessed to pick out publications for each PI in order to update their publication lists on their personal webpages.

Specific publications for PI website

- 1) The database is directly accessed in order to populate the relevant PIs personal website with their publications.

This step directly involves interacting with the database and returning a query of articles from a specific PI in order to populate that PI's website with their publications. From this use case, the following entities can be identified: **researcher, paper, journal**. These three entities are crucial in order to identify the correct papers that will be populating the websites. These entities will be used to form the main relationships in the database. Using the following business rules and relationships, the correct papers can be correctly assigned to the PI's website.

Structural Rule 1: Each paper is associated with one or many researchers; Each researcher is associated with one or many papers

Structural Rule 2: A researcher is a LCN faculty, LCN member, LCN collaborator or none of these.

For Structural rule one, this rule is necessary because at times a paper is written by one person and at other times there are several people who collaborate on a paper (plural). This speaks to the participation involved in this rule, where there is an option for one or more than one researcher to be involved in authoring a paper. Structural rule two is a specialization for the researcher entity and has a partially complete disjoint constraint. This means that not all researchers in the database will fall under the LCN_faculty, LCN_member and LCN_collaborator subtypes and that a researcher cannot be a part of more than one subtype.

Structural Rule 3: Each paper is associated with a journal; A journal is associated with one or many papers

Structural rule three states that a paper is only associated with one journal because it is not possible to publish a paper in several journals (singular). Papers cannot even be submitted to several journals at one time for review. If you wish to submit a paper to another journal then it must first be retracted from the first and then submitted to the second journal. For this reason there is no optionality in the number of associated participations. Again a journal is a large organization so therefore it is associated with many papers (plural). The fields from use case one are applicable to this use case.

Use case 3

- The user accesses the database through the website application in order to sort publications by year (ascending and descending), citation number (ascending and descending) journal impact factor (ascending and descending) and first author last name. The website will also be able to provide a list of papers depending on the field (subject area) of the journals and specific topics that LCN does research on. This will require adding two new entities to the database, **Journal_field** and **Paper_topic**.
- 1) The person visit's the PI's website and interacts with the sort button on the website.
 - 2) The database is accessed in order to return a query with the sorted list requested by the end user.

Structural Rule 4: Each journal field is associated with a journal; Each journal is associated with a journal field

Structural Rule 5: Each paper topic is associated with one or many papers; Each paper is associated with one of many paper topics

Step 1) entails the user interacting with the website, in step 2) the website directly interacts with the database in order to produce the user's expected result. Many of the structural rules, relationships and fields from use case 2 are applicable to use cause 3 specifically when the user is sorting by year, citation, first and last name. The author first name and last name are both attributes under the researcher entity, and are therefore subject to related structural rules.

For structural rule four there is no option in participation since each journal only publishes papers from a specific topic. Therefore each journal can only have a topic and there can only be one (singular). For structural rule five, since the topics are specific to projects that LCN chooses to focus on, a paper can span several topics (plural). There is also mandatory participation, since the nature of publishing a paper requires that it be about a specific topic.

| Field | What it Stores | Why it's Needed |
|--------------------|---|--|
| Journal_field_name | This is the subject area of the journal where the paper is published. | This will help the end user to sort through papers, eg if they are only interested in Neuroscience papers then they can only bring those up. |
| Paper_topic_name | This will house the specific research topics that LCN works on. | This will help the end user to sort through papers, eg if they are only interested in optical imaging papers from LCN. |

Use case 4

- Users can search the database using the search bar on the website to identify a publication of interest by publication name, search for papers published in a specific journal or by a specific researcher.
- 1) The user visit's the PI's website and interacts with the search bar.
 - 2) The website application then interacts with the database in order to return the relevant publications.

Step 1) entails the user interacting with the website, in step 2) the website directly interacts with the database in order to produce the user's expected result. The structural rules and fields for this use case have already been established under use case 2.

Use case 5

- Users can access the PDF papers once a paper of interest is selected on the website.
- 1) The user visits the PI's website and selects a paper of interest.
 - 2) If the paper is an open source paper then the user can have free access to the PDF.
 - 3) If the paper is from a journal that has a pay wall, and if the user belongs to LCN they can create an account and enter their password to access the PDF of the paper.

Structural Rule 6: A paper PDF is a Paywall paper or an Open source paper.

Structural Rule 7: Each payroll paper may be associated with one or many accounts; An account may be associated with one or many payroll papers

Step 1) involves the user interacting with the website to select a paper, the website sources the database in order to produce the paper PDF in step 2). If the user is an LCN member and they are trying to access a paper that is payroll restricted. In step 3) the user must create an account and password, which is stored in the database, in order to access their paper of interest.

Structural rule six is a specialization of the paper entity, where PDFs of the paper in the database can be stored. The subtypes are Paywall_paper and Opensource_paper and the relationship is complete with a disjoint constraint. This means that all papers must be assigned to one of the two subtypes and they can only be in one subtype at a time.

For structural rule seven there is optionality in the relationship between a paper PDF and an account. A user with an account can access no PDFs, one or many payroll PDFs (plural) and similarly a PDF may be accessed by no accounts, one or many accounts.

Use case 6

- Use of a trigger to maintain the history tables of the citation numbers of papers and the impact factors of journals.
- 1) The citation number of a paper or the impact factor of a journal changes and is updated in the database.
 - 2) The track historical changes trigger will be activated and add the old value, new value and foreign key to the relevant historical table.

Structural Rule 8: Each journal is associated with one or many impact factors; Each impact factor is associated with one journal

Structural Rule 9: Each paper is associated with one or many paper citation numbers; Each paper citation number is associated with one paper

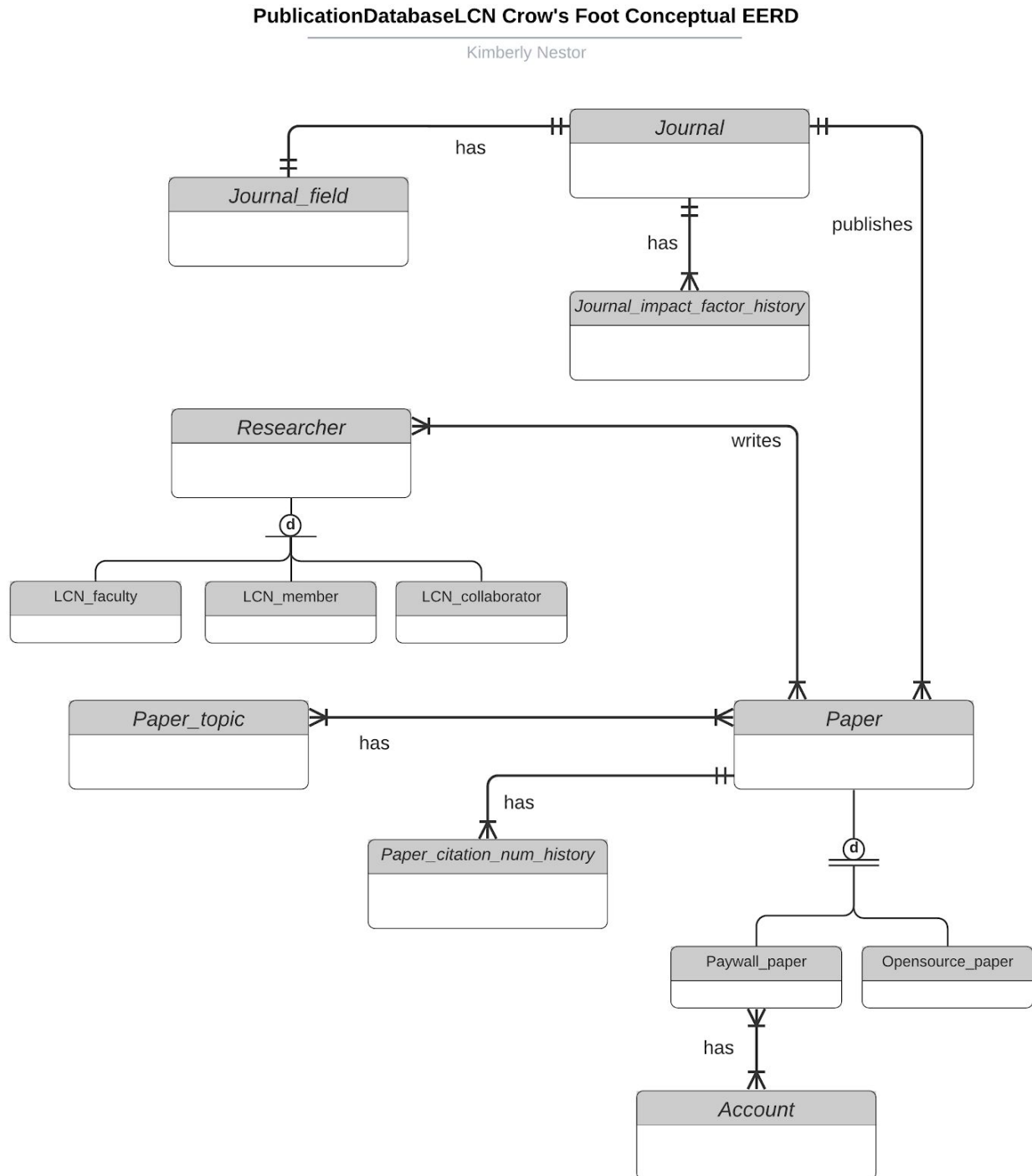
| Field | What it Stores | Why it's Needed |
|-------------------|---|--|
| Old_impact_factor | This is the former impact factor of the journal. | This will allow the end user to compare former impact factors with the current impact factor. |
| New_impact_factor | This is the current impact factor of the journal. | This will tell the end user what the new impact factor is. |
| Old_citation_num | This is the former citation number of the journal. | This will allow the end user to compare former citation numbers with the current citation number. |
| New_citation_num | This is the current citation number of the journal. | This will tell the end user what the new citation number is. |
| Change_date | This is the date either the impact fact of the citation number changed. | This will allow the user to determine when the change was made and do data analysis on the citation numbers or impact factors. |

PublicationDatabaseLCN EERD

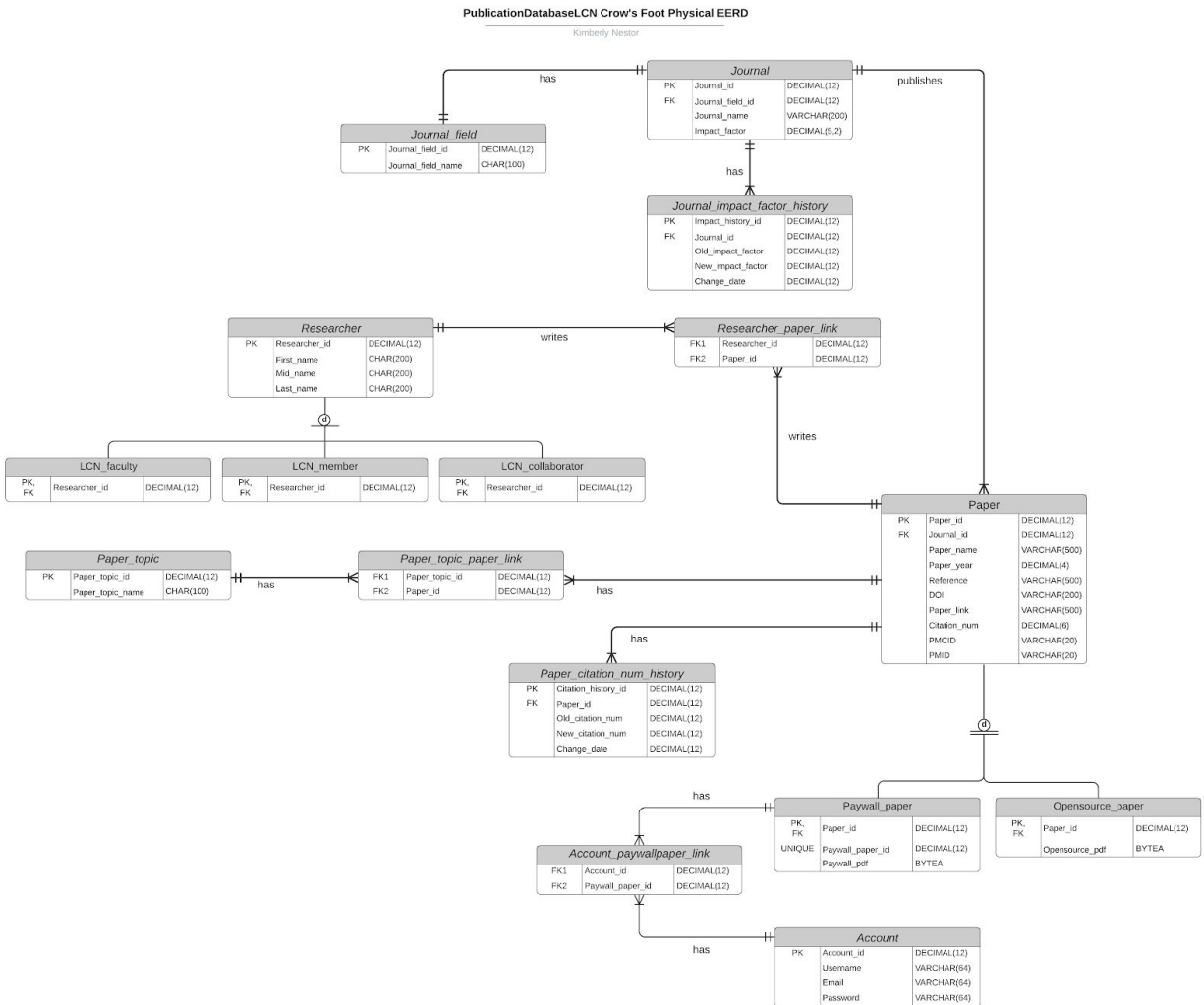
Here is a concise summary of the structural rules I came up with for my database plan.

1. Each paper is associated with one or many researchers; Each researcher is associated with one or many papers
2. A researcher is a LCN faculty, LCN member, LCN collaborator or none of these.
3. Each paper is associated with a journal; A journal is associated with one or many papers
4. Each journal field is associated with a journal; Each journal is associated with a journal field
5. Each paper topic is associated with one or many papers; Each paper is associated with one of many paper topics
6. A paper PDF is a Paywall paper or an Open source paper.
7. Each paywall paper may be associated with one or many accounts; An account may be associated with one or many paywall papers
8. Each journal is associated with one or many impact factors; Each impact factor is associated with one journal
9. Each paper is associated with one or many paper citation numbers; Each paper citation number is associated with one paper

Below is the conceptual EERD I created to model the structural rules above using Crow's foot notation.



Below is the resulting physical EERD, based on the conceptual EERD, I created to model the structural rules using Crow's foot notation.



Based on the EERD there are two many to many relationships; paper and paper topic, paywall paper and account. The other associative relationships in the database are one to one and one to many relationships with journal and journal field, journal and paper respectively. Based on the diagram, both journal field and paper name are only capable of being associated with one journal, while a journal can be associated with one field and many papers. So far all the associative relationships in the database are mandatory and there is no relationship that does not have at least one participant.

Other notable relationships in the database are specialization relationships. The researcher supertype and the subsequent subtypes, LCN_faculty, LCN_member and LCN_collaborator will

make it easier to create websites for faculty using the database later on. Users of the website can also sort for papers using the specialized relationships. The other specialization relationship in the EERD is the Paper supertype and the subtypes Paywall_paper and Opensource_paper. The purpose of this relationship is to restrict the access of papers from journals that have a paywall from the general public. Only LCN lab members with an account will be able to access the paywall papers.

There are two history tables in the database maintained by triggers, Journal_impact_factor_history and Paper_citation_num_history. These tables track the changes of the impact factors of journals and the citation numbers of papers. They are automatically updated when the changes are made due to the triggers. There is a one to many relationship in both tables with the Journal and Paper entities because a Journal or Paper may have many impact factors or citation numbers over their lifetime, but each value for both entities can only be related to that specific entity.

PublicationDatabaseLCN Normalization

It is my opinion that the current physical EERD I created has normalized the database to the fullest extent that is needed for this project. The following entities in the database are in BCNF: Journal_field, Researcher_paper_link, Paper_topic, Paper_topic_paper_link, LCN_faculty, LCN_member, LCN_collaborator, Paywall_paper, Opensource_paper and Account_paywallpaper_link. The remaining entities Researcher, Paper and Account are not in BCNF, but are in 3NF. I am currently opting to leave these entities in 3NF and not normalize them to BCNF because I do not think this level of normalization is necessary for the purposes of this database. The extra tables that will be created from this level of normalization will only be a hindrance to the database instead of a benefit. There will be a small number of end users who use this database and therefore, does not warrant BCNF level of normalization.

PublicationDatabaseLCN SQL Script

Here is a screenshot of the execution of the SQL script I used to create all the tables in my database, based on the entities in the physical EERD above. Please see the other attachment for the full SQL script.

The screenshot shows a PostgreSQL Query Editor window with the following SQL commands:

```

14 DROP TABLE Journal;
15 DROP TABLE Journal_field;
16
17
18 CREATE TABLE Journal_field (
19     Journal_field_id DECIMAL(12) NOT NULL PRIMARY KEY,
20     Journal_field_name CHAR(100) NOT NULL );
21
22 CREATE TABLE Journal (
23     Journal_id DECIMAL(12) NOT NULL PRIMARY KEY,
24     Journal_field_id DECIMAL(12) NOT NULL,
25     Journal_name VARCHAR(200) NOT NULL,
26     Impact_factor DECIMAL(3,2),
27     FOREIGN KEY (Journal_field_id) REFERENCES Journal_field(Journal_field_id));
28
29 CREATE TABLE Paper (
30     Paper_id DECIMAL(12) NOT NULL PRIMARY KEY,
31     Journal_id DECIMAL(12) NOT NULL,
32     Paper_name VARCHAR(500) NOT NULL,
33     Paper_year DECIMAL(4) NOT NULL,
34     Reference VARCHAR(500) NOT NULL,
35     DOI VARCHAR(200),
36     Paper_link VARCHAR(500) NOT NULL,
37     Citation_num DECIMAL(6) NOT NULL,
38     PMCID VARCHAR(20),
39     PMID DECIMAL(20),
40     FOREIGN KEY (Journal_id) REFERENCES Journal(Journal_id));
41

```

The Messages pane at the bottom shows the following output:

```

CREATE TABLE
Query returned successfully in 62 msec.

```

A green status bar at the bottom right indicates: "Query returned successfully in 62 msec."

PublicationDatabaseLCN Indexing

The first set of indexes for this database are the primary keys, which is mandatory and will be implemented by the DBMS. The following is a list of primary keys for this database:

- Journal_field.Journal_field_id
- Journal.Journal_id
- Researcher.Researcher_id
- LCN_faculty.Researcher_id
- LCN_member.Researcher_id
- LCN_collaborator.Researcher_id
- Paper.Paper_id
- Paper_topic.Paper_topic_id
- Opensource_paper.Paper_id
- Paywall_paper.Paper_id
- Account.Account_id

Since all foreign keys need an index, the following is a table of all the foreign keys in the database and information on whether they are unique:

| Column | Unique Constraint | Description |
|---------------------------------------|-------------------|--|
| Journal.Journal_field_id | Not unique | The foreign key in Journal referencing the journal field is not unique because there may be many journals in a particular field. |
| Researcher_paper_link.Paper_id | Not Unique | The foreign key in Researcher_paper_link referencing Paper_id is unique because while each paper is unique there may be many researchers who coauthor a paper so one Paper_id will be referenced several times in the table. |
| Researcher_paper_link.Researcher_id | Not unique | The foreign key in Researcher_paper_link referencing Researcher_id is not unique because a researcher can be an author on several papers. |
| LCN_faculty.Researcher_id | Unique | The foreign key in LCN_faculty referencing Researcher_id is unique because there is only one version of every researcher since they are a person. |
| LCN_member.Researcher_id | Unique | The foreign key in LCN_member referencing Researcher_id is unique because there is only one version of every researcher since they are a person. |
| LCN_collaborator.Researcher_id | Unique | The foreign key in LCN_collaborator referencing Researcher_id is unique because there is only one version of every researcher since they are a person. |
| Paper_topic_paper_link.Paper_topic_id | Not unique | The foreign key in Paper_topic_paper_link referencing Paper_topic_id is not unique because |

| | | |
|--|------------|--|
| | | a paper topic may be referenced several times as many papers can have the same topic. |
| Paper_topic_paper_link.Paper_id | Not unique | The foreign key in Paper_topic_paper_link referencing Paper_id is not unique because a paper may have many topics and the ID would therefore need to be repeated for each topic. |
| Paper.Journal_id | Not Unique | The foreign key in Paper referencing Journal_id is not unique because a journal can publish many papers. |
| Paywall_paper.Paper_id | Unique | The foreign key in Paywall_paper referencing Paper_id is unique because each paper can only have one PDF. |
| Opensource_paper.Paper_id | Unique | The foreign key in Opensource_paper referencing Paper_id is unique because each paper can only have one PDF. |
| Account_paywallpaper_link.Account_id | Not unique | The foreign key in Account_paywallpaper_link referencing Account_id is not unique because an account can access several paper PDFs. |
| Account_paywallpaper_link.Paywall_paper_id | Not unique | The foreign key in Account_paywallpaper_link referencing Paywall_paper_id is not unique because a paper PDF can be accessed by several accounts. |

I selected the following for my three query driven indexes; Paper.Paper_year, Paper.Citation_num and Journal.Impact_factor. The Paper_year entity can be used in a query using the ORDER BY command, to return a list of papers sorted in ascending or descending order by year. Paper_year can also be used in a query using the WHERE or HAVING command to return papers from specific years. This query can be further specified to return papers from a specific year published by a specific researcher. The Citation_num will also be a popular way to sort papers using the ORDER BY command. The number of citations is a numeric value that is used to see how popular a paper is and directly relates to the number of times a paper is

referenced in other peer reviewed articles. Comparison operators can be used along with WHERE and HAVING statements to return papers within a certain citation range. Similarly to the citation_num the Impact_factor is a value assigned to journals that is determined by the number of citations papers published in that journal have amassed. This number allows readers to determine the prominence of a journal. Similarly, WHERE, HAVING and ORDER BY statements can be used on the Impact_factor. All three can be used as means to restrict the returned results in a query and can greatly benefit from indexing to ensure the query return time is quick:

| Column | Unique Constraint | Description |
|-----------------------|-------------------|---|
| Paper.Paper_year | Not unique | The foreign key in Paper referencing Paper_year is not unique because there can be several papers published in one year. |
| Paper.Citation_num | Not unique | The foreign key in Paper referencing Citation_num is not unique because though it is less likely, several papers can have the same number of citations. |
| Journal.Impact_factor | Not unique | The foreign key in Journal referencing Impact_factor is not unique because though it is less likely, several journals can have the same impact factor. |

Here is a screenshot of the execution of my index commands. Please see the SQL script for the full thing.

CS669/postgres@PostgreSQL 12

Query Editor Query History

```
100 --Foreign Key Non-Unique INDEXES
101 CREATE INDEX JournalFieldIdx
102 ON Journal(Journal_field_id);
103
104 CREATE INDEX ResPapLinkPaperIdx
105 ON Researcher_paper_link(Paper_id);
106
107 CREATE INDEX ResPapLinkResearcherIdx
108 ON Researcher_paper_link(Researcher_id);
109
110 CREATE INDEX PapTopPapLinkPapTopIdx
111 ON Paper_topic_paper_link(Paper_topic_id);
112
113 CREATE INDEX PapTopPapLinkPapIdx
114 ON Paper_topic_paper_link(Paper_id);
115
116 CREATE INDEX PaperJournalIdx
117 ON Paper(Journal_id);
118
119 CREATE INDEX AccPaywallPapLinkAccIdx
120 ON Account_paywallpaper_link(Account_id);
121
122 CREATE INDEX AccPaywallPapLinkPaywallPapIdx
123 ON Account_paywallpaper_link(Paywall_paper_id);
124
125
126 --Foreign Key Unique INDEXES
```

Scratch Pad

Non-Unique Index

```
CREATE INDEX IndexName
ON TableName (ColumnList);

CREATE INDEX CarPriceldx
ON Car(Price);
```

Unique Index

```
CREATE UNIQUE INDEX IndexName
ON TableName (ColumnList);

CREATE UNIQUE INDEX CarVINIdx
ON Car(VIN);
```

Messages Data Output Explain Notifications

CREATE INDEX

Query returned successfully in 78 msec.

✓ Query returned successfully in 78 msec.

PublicationDatabaseLCN Transactions

The following three use cases are relevant to the transactions I created:

Use case 2

- Database is accessed to pick out **publications** for each PI in order to update their publication lists on their personal webpages.

Specific publications for PI website

- 2) The database is directly accessed in order to populate the relevant PIs personal website with their publications.

Use case 3

- The user accesses the database through the website application in order to sort publications by **year** (ascending and descending), **citation number** (ascending and descending) **journal impact factor** (ascending and descending) and first author **last name**. The website will also be able to provide a list of papers depending on the field (subject area) of the journals and specific topics that LCN does research on. This will require adding two new entities to the database, Journal_field and Paper_topic.
- 3) The person visit's the PI's website and interacts with the sort button on the website.
 - 4) The database is accessed in order to return a query with the sorted list requested by the end user.

Use case 4

- Users can search the database using the search bar on the website to identify a publication of interest by **publication name**, search for papers published in a specific **journal** or by a specific **researcher**.
- 3) The user visit's the PI's website and interacts with the search bar.
 - 4) The website application then interacts with the database in order to return the relevant publications.

Transaction 1

For this transaction I created the procedure "AddJournal", which is relevant to use case three. In this transaction the journals in which papers are published is entered into the database. The inputs for the procedure are Journal_id, Journal_field_id, Journal_name and Impact_factor. The Journal_field_id is a foreign key that links the journal to the subject area that it is in. All inputs to the procedure are relevant attributes in the Journal table. The INSERT INTO, VALUES command is the prompt in the procedure to insert the field information into the Journal table.

```

177
178 --AddJournal function
179 CREATE OR REPLACE FUNCTION AddJournal(Journal_id IN DECIMAL, Journal_field_id IN DECIMAL,
180                                     Journal_name IN VARCHAR, Impact_factor IN DECIMAL)
181 RETURNS VOID
182 AS
183 $proc$
184 BEGIN
185     INSERT INTO Journal(Journal_id, Journal_field_id, Journal_name, Impact_factor)
186     VALUES(Journal_id, Journal_field_id, Journal_name, Impact_factor);
187
188     /*
189     --make journal subtypes later
190     INSERT INTO Peer_review?(Journal_id)
191     VALUES(Journal_id);
192
193     INSERT INTO Conference_proceedings(Journal_id)
194     VALUES(Journal_id);
195     */
196
197 END;
198 $proc$ LANGUAGE plpgsql;
199
200
201
202
203
204
205
206

```

Messages Explain Notifications Query History

Successfully run. Total query runtime: 89 msec.
26 rows affected.

In the below screenshot the stored procedure “AddJournal” is executed, to add the journals Acta Neuropathologica, Advances in Alzheimer’s Disease and American Journal of Neuroradiology to the Journal table. The successful addition of the Journal_name and other related attributes can be seen in the data output section. I used three DO, BEGIN and EXECUTE commands to add the Journal information.

```

197
198 END;
199 $proc$ LANGUAGE plpgsql;
200
201 --AddJournal Execute function
202 START TRANSACTION;
203 DO
204 $$ BEGIN
205     EXECUTE AddJournal(11001, 10507, 'Acta Neuropathologica', 18.174);
206     END$$;
207 --COMMIT TRANSACTION;
208
209 DO
210 $$ BEGIN
211     EXECUTE AddJournal(11002, 10504, 'Advances in Alzheimer's Disease', 2.13);
212     END$$;
213
214 DO
215 $$ BEGIN
216     EXECUTE AddJournal(11003, 10506, 'American Journal of Neuroradiology', 3.653);
217     END$$;
218
219 SELECT journal_name, journal_field_name, impact_factor
220 FROM Journal
221 JOIN Journal_field ON Journal_field.Journal_field_id = Journal.Journal_field_id
222

```

Messages Explain Notifications Query History Data Output

| | journal_name character varying (200) | journal_field_name character (100) | impact_factor numeric (5,3) |
|---|---|---------------------------------------|--------------------------------|
| 1 | Acta Neuropathologica | Neuroscience | 18.174 |
| 2 | Advances in Alzheimer's Disease | Health Professions | 2.130 |
| 3 | American Journal of Neuroradiology | Medicine | 3.653 |

Transaction 2

For transaction two I created the procedure "AddResearcher", which is relevant to use case four. In this transaction the researchers who are authors on papers that will be stored in the database are added to the Researcher table. The inputs to the procedure are Researcher_id, First_name, Mid_name, Last_name, and Researcher_type. The first three inputs are attributes in the Researcher table, however the last input Researcher_type is not. The Researcher_type input is necessary because the entity Researcher has three partially complete disjoint subtypes, LCN_faculty, LCN_member and LCN_collaborator. The screenshot below shows the code for and this stored procedure.

Within the BEGIN command there are four IF statements that are directly related to the Researcher_type input. These statements are necessary to determine which subtype the researcher is assigned, or a researcher can be assigned to none since the subtype relationship is partially complete. The input for Researcher_type is a string of either 'faculty', 'member', 'collaborator' or ''. Based on the character input the researcher and associated attributes will be inserted into the Researcher table and the Researcher_id is inserted into the appropriate subtype table. The last input option for Researcher_type is a null value ('') which will indicate that a researcher is not a part of any subtype and only an insert into the Researcher table occurs.

```

236 --AddResearcher function
237 CREATE OR REPLACE FUNCTION AddResearcher(Researcher_id IN DECIMAL, First_name IN CHAR,
238                                         Mid_name IN CHAR, Last_name IN CHAR, Researcher_type IN CHAR)
239 RETURNS VOID
240 AS
241 $proc$
242 BEGIN
243     IF Researcher_type = 'faculty' THEN
244         INSERT INTO Researcher(Researcher_id, First_name, Mid_name, Last_name)
245         VALUES(Researcher_id, First_name, Mid_name, Last_name);
246         INSERT INTO LCN_faculty(Researcher_id)
247         VALUES(Researcher_id);
248     END IF;
249     IF Researcher_type = 'member' THEN
250         INSERT INTO Researcher(Researcher_id, First_name, Mid_name, Last_name)
251         VALUES(Researcher_id, First_name, Mid_name, Last_name);
252         INSERT INTO LCN_member(Researcher_id)
253         VALUES(Researcher_id);
254     END IF;
255     IF Researcher_type = 'collaborator' THEN
256         INSERT INTO Researcher(Researcher_id, First_name, Mid_name, Last_name)
257         VALUES(Researcher_id, First_name, Mid_name, Last_name);
258         INSERT INTO LCN_collaborator(Researcher_id)
259         VALUES(Researcher_id);
260     END IF;
261     IF Researcher_type = '' THEN
262         INSERT INTO Researcher(Researcher_id, First_name, Mid_name, Last_name)
263         VALUES(Researcher_id, First_name, Mid_name, Last_name);
264     END IF;
265 END;
266 $proc$ LANGUAGE plpgsql;
267

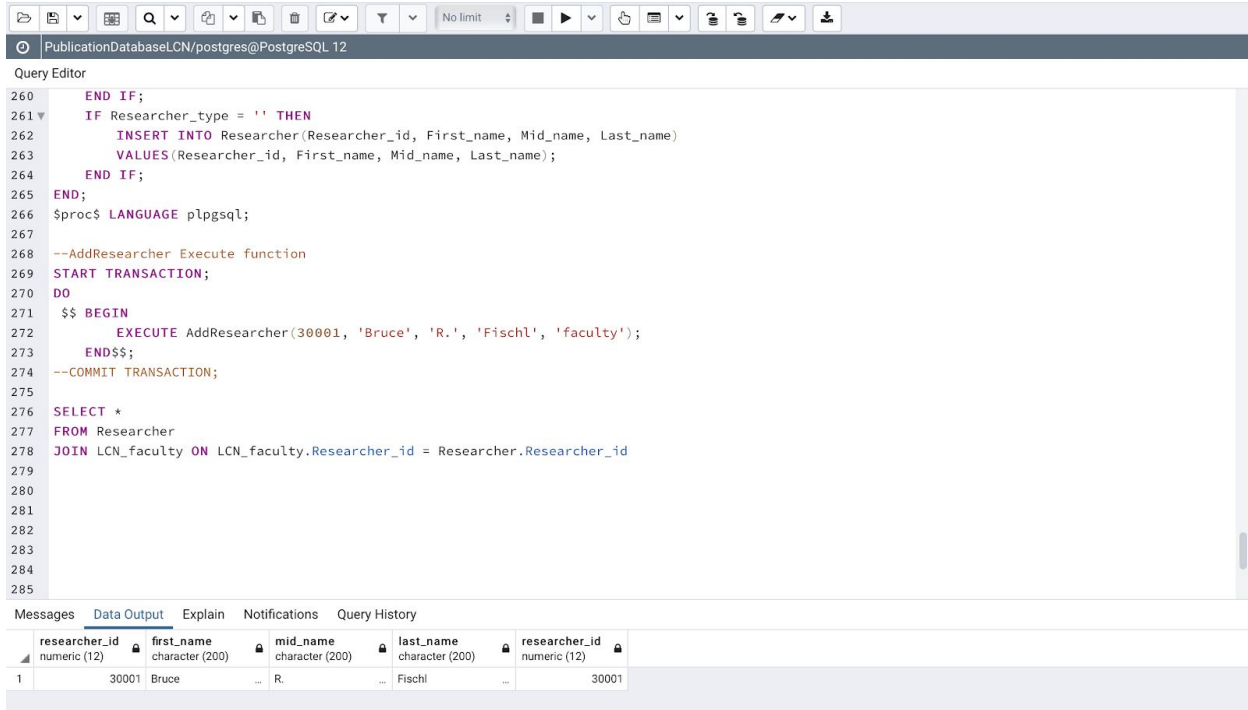
```

Messages Data Output Explain Notifications Query History

WARNING: there is already a transaction in progress

Successfully run. Total query runtime: 85 msec.
1 rows affected.

The screenshot below shows the execution of the AddResearcher stored procedure and the effective use of the IF statements into procedural SQL block. In the screenshot, the researcher Bruce Fischl is used as an example to insert attribute information into the entities Researcher and LCN_faculty. Both primary keys are shown to demonstrate that the information was added to both tables since a join was used to ensure only one screenshot is needed. The other three IF statements ('member', 'collaborator', and ") were also tested using the same researcher attributes to ensure the stored procedure worked effectively. Screenshots are not shown to conserve on space.



```

260 END IF;
261 IF Researcher_type = ' ' THEN
262     INSERT INTO Researcher(Researcher_id, First_name, Mid_name, Last_name)
263     VALUES(Researcher_id, First_name, Mid_name, Last_name);
264 END IF;
265 END;
266 $proc$ LANGUAGE plpgsql;
267
268 --AddResearcher Execute function
269 START TRANSACTION;
270 DO
271 $$ BEGIN
272     EXECUTE AddResearcher(30001, 'Bruce', 'R.', 'Fischl', 'faculty');
273 END$$;
274 --COMMIT TRANSACTION;
275
276 SELECT *
277 FROM Researcher
278 JOIN LCN_faculty ON LCN_faculty.Researcher_id = Researcher.Researcher_id
279
280
281
282
283
284
285

```

| | researcher_id numeric (12) | first_name character (200) | mid_name character (200) | last_name character (200) | researcher_id numeric (12) |
|---|-------------------------------|-------------------------------|-----------------------------|------------------------------|-------------------------------|
| 1 | 30001 | Bruce | R. | Fischl | 30001 |

Transaction 3

For transaction three I created the procedure "AddPaper", which is relevant to use cases two, three and four. In this transaction papers that are published by researchers in LCN will be added to the Paper entity in the database. The inputs to the procedure are Paper_id, Journal_id, Paper_name, Paper_year, Reference, DOI, Paper_link, Citation_num, PMCID, and PMID.

The screenshot below shows the code for and the effective execution of the AddPaper stored procedure. Since PMCID and PMID are nullable values they were entered as nulls into the stored procedure since they are not the most relevant attributes of a paper. These values can be added later with an UPDATE command. The INSERT INTO command in the stored procedure adds the paper information to the Paper entity, which is executed by the EXECUTE command.

```

604
605 --AddPaper function
606 CREATE OR REPLACE FUNCTION AddPaper(Paper_id IN DECIMAL, Journal_id IN DECIMAL, Paper_name IN VARCHAR,
607                                     Paper_year IN DECIMAL, Reference IN VARCHAR, DOI IN VARCHAR, Paper_link IN VARCHAR,
608                                     Citation_num IN DECIMAL, PMCID IN VARCHAR, PMID IN VARCHAR)
609 RETURNS VOID
610 AS
611 $proc$
612 BEGIN
613     INSERT INTO Paper(Paper_id, Journal_id, Paper_name, Paper_year, Reference, DOI, Paper_link, Citation_num, PMCID, PMID)
614     VALUES(Paper_id, Journal_id, Paper_name, Paper_year, Reference, DOI, Paper_link, Citation_num, PMCID, PMID);
615
616 END;
617 $proc$ LANGUAGE plpgsql;
618
619 --AddPaper Execute function
620
621 START TRANSACTION;
622 DO
623 $$ BEGIN
624     EXECUTE AddPaper(20001, 11026, 'Multimodal image registration through simultaneous segmentation',
625                     2017, 'Aganj, I., Fischl, B. (2017).Multimodal image registration through simultaneous segmentation.
626                     IEEE Signal Processing Letters,24(11) 1661-1665', '10.1109/LSP.2017.2754263',
627                     'https://www.scopus.com/record/display.url?eid=2-s2.0-85030764251&origin=resultslist', 2, '', '');
628
629 END$$;
630
631 SELECT Paper_name, Paper_year, Reference, DOI, Paper_link, Citation_num
632 FROM Paper
633

```

Messages Explain Notifications Query History

WARNING: there is already a transaction in progress
 WARNING: there is already a transaction in progress

Successfully run. Total query runtime: 96 msec.
 1 rows affected.

The screenshot below shows the data output of the Paper table, showing the successful addition of the paper added using the AddPaper stored procedure.

| | paper_name character varying (500) | paper_year numeric (4) | reference character varying (500) | doi character varying (200) | paper_link character varying (500) | citation_num numeric (6) |
|---|--|---------------------------|--|--------------------------------|---------------------------------------|-----------------------------|
| 1 | Multimodal image registration through simulta... | 2017 | Aganj, I., Fischl, B. (2017).Multimodal i... | 10.1109/LSP.2017.2754263 | https://www.scopus.com/rec... | 2 |

Messages Explain Notifications Query History

WARNING: there is already a transaction in progress
 WARNING: there is already a transaction in progress

Successfully run. Total query runtime: 96 msec.
 1 rows affected.

PublicationDatabaseLCN Triggers

Trigger 1 - Journal Impact factor history

This trigger will record changes made to the impact factor of a journal. The trigger will log the journal_id, the old impact factor and the new impact factor into a table called Journal_impact_factor_history. This table can be accessed to see the historical view of the impact factors of journals. The trigger is activated when there is a new insert or an update of an Impact_factor in the Journal table. The screenshot below shows the effective creation of the trigger.

```

1450 --Journal impact factor history Trigger
1451 CREATE TABLE Journal_impact_factor_history (
1452   Impact_history_id DECIMAL(12) NOT NULL PRIMARY KEY,
1453   Journal_id DECIMAL(12) NOT NULL,
1454   Old_impact_factor DECIMAL(5,3) NOT NULL,
1455   New_impact_factor DECIMAL(5,3) NOT NULL,
1456   Change_date DATE NOT NULL,
1457   FOREIGN KEY (Journal_id) REFERENCES Journal(Journal_id));
1458
1459
1460 --trigger
1461 CREATE OR REPLACE FUNCTION Journal_impact_factor_history_func() RETURNS TRIGGER LANGUAGE plpgsql
1462 AS $triggerfunc$
1463 BEGIN
1464 IF OLD.Impact_factor != NEW.Impact_factor THEN
1465   INSERT INTO Journal_impact_factor_history(Impact_history_id, Journal_id,
1466     Old_impact_factor, New_impact_factor, Change_date)
1467   VALUES(COALESCE((SELECT MAX(Impact_history_id)+1 FROM Journal_impact_factor_history), 50000), NEW.Journal_id,
1468     OLD.Impact_factor, NEW.Impact_factor, current_date);
1469   END IF;
1470   RETURN NEW;
1471 END;
1472 $triggerfunc$;
1473 CREATE TRIGGER Journal_impact_factor_history_trig
1474 BEFORE UPDATE OF Impact_factor ON Journal
1475 FOR EACH ROW
1476 EXECUTE PROCEDURE Journal_impact_factor_history_func();
1477

```

Messages Query History Notifications Explain

WARNING: there is already a transaction in progress
CREATE TRIGGER

Query returned successfully in 137 msec.

The screenshot below shows that the Journal_impact_factor_history_trig works effectively when an update is made on the Impact_factor in Journal. There is now a row in the table that logged the old and new impact factors as well as the change date.

```

1460 --trigger
1461 CREATE OR REPLACE FUNCTION Journal_impact_factor_history_func() RETURNS TRIGGER LANGUAGE plpgsql
1462 AS $triggerfunc$
1463 BEGIN
1464 IF OLD.Impact_factor != NEW.Impact_factor THEN
1465   INSERT INTO Journal_impact_factor_history(Impact_history_id, Journal_id,
1466     Old_impact_factor, New_impact_factor, Change_date)
1467   VALUES(COALESCE((SELECT MAX(Impact_history_id)+1 FROM Journal_impact_factor_history), 50001), NEW.Journal_id,
1468     OLD.Impact_factor, NEW.Impact_factor, current_date);
1469   END IF;
1470   RETURN NEW;
1471 END;
1472 $triggerfunc$;
1473 CREATE TRIGGER Journal_impact_factor_history_trig
1474 BEFORE UPDATE OF Impact_factor ON Journal
1475 FOR EACH ROW
1476 EXECUTE PROCEDURE Journal_impact_factor_history_func();
1477
1478
1479 --INSERT INTO Journal VALUES(11002, 10504, 'Advances in Alzheimer's Disease', 2.13);
1480 --test trigger ~ journal impact factor
1481 UPDATE Journal
1482 SET Impact_factor = 5.44
1483 WHERE Journal_id = 11002;
1484
1485 SELECT *
1486 FROM Journal_impact_factor_history;
1487

```

Messages Query History Notifications Explain Data Output

| impact_history_id [PK] numeric(12) | journal_id numeric(12) | old_impact_factor numeric(5,3) | new_impact_factor numeric(5,3) | change_date date |
|---------------------------------------|---------------------------|-----------------------------------|-----------------------------------|---------------------|
| 1 | 80001 | 11002 | 2.130 | 5.440 2020-06-23 |

Trigger 2 - Paper Citation number history

This trigger will record changes made to the citation number of a paper. The trigger will log the paper_id, the old citation number and the new citation number into a table called Paper_citation_num_history. This table can be accessed to see the historical view of the citation number of papers. The trigger is activated when there is a new insert or an update of a citation_num in the Paper table. The screenshot below shows the effective creation of the trigger.

```

1478
1479 --Paper citation number history Trigger
1480 CREATE TABLE Paper_citation_num_history (
1481     Citation_history_id DECIMAL(12) NOT NULL PRIMARY KEY,
1482     Paper_id DECIMAL(12) NOT NULL,
1483     Old_citation_num DECIMAL(6) NOT NULL,
1484     New_citation_num DECIMAL(6) NOT NULL,
1485     Change_date DATE NOT NULL,
1486     FOREIGN KEY (Paper_id) REFERENCES Paper(Paper_id));
1487
1488 --trigger
1489 CREATE OR REPLACE FUNCTION Paper_citation_num_history_func() RETURNS TRIGGER LANGUAGE plpgsql
1490 AS $trigfunc$
1491 BEGIN
1492     IF OLD.Citation_num != NEW.Citation_num THEN
1493         INSERT INTO Paper_citation_num_history(Citation_history_id, Paper_id,
1494             Old_citation_num, New_citation_num, Change_date)
1495             VALUES(COALESCE((SELECT MAX(Citation_history_id)+1 FROM Paper_citation_num_history), 40000), NEW.Paper_id,
1496                 OLD.Citation_num, NEW.Citation_num, current_date);
1497     END IF;
1498     RETURN NEW;
1499 END;
1500 $trigfunc$;
1501 CREATE TRIGGER Paper_citation_num_history_trig
1502 BEFORE UPDATE OF Citation_num ON Paper
1503 FOR EACH ROW
1504 EXECUTE PROCEDURE Paper_citation_num_history_func();
1505
1506

```

Messages Query History Notifications Explain

WARNING: there is already a transaction in progress
CREATE TRIGGER

Query returned successfully in 137 msec.

The screenshot below shows that the Paper_citation_num_history_trig works effectively when an update is made on the Citation_num in Paper. There is now a row in the table that logged the old and new citation numbers as well as the change date.

```

1508         OLD.Citation_num, NEW.Citation_num, current_date);
1509     END IF;
1510     RETURN NEW;
1511 END;
1512 $trigfunc$;
1513 CREATE TRIGGER Paper_citation_num_history_trig
1514 BEFORE UPDATE OF Citation_num ON Paper
1515 FOR EACH ROW
1516 EXECUTE PROCEDURE Paper_citation_num_history_func();
1517
1518 /*
1519 INSERT INTO Paper VALUES(20001, 11001, 'Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and
1520     frontotemporal dementia', 2017, 'Yokoyama, J.S., Karch, C.M., Fan, C.C. and 28 more (...)
1521     (2017).Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy,
1522     and frontotemporal dementia. Acta Neuropathologica,133(5) 825-837', '10.1007/s00401-017-1693-y',
1523     'https://www.scopus.com/record/display.url?eid=2-s2.0-85014531490&origin=resultslist', 31, '', '');
1524 */
1525 --test trigger - paper citation number
1526 UPDATE Paper
1527 SET Citation_num = 46
1528 WHERE Paper_id = 20001;
1529
1530 SELECT *
1531 FROM Paper_citation_num_history;
1532
1533
1534
1535
1536

```

Messages Query History Notifications Explain Data Output

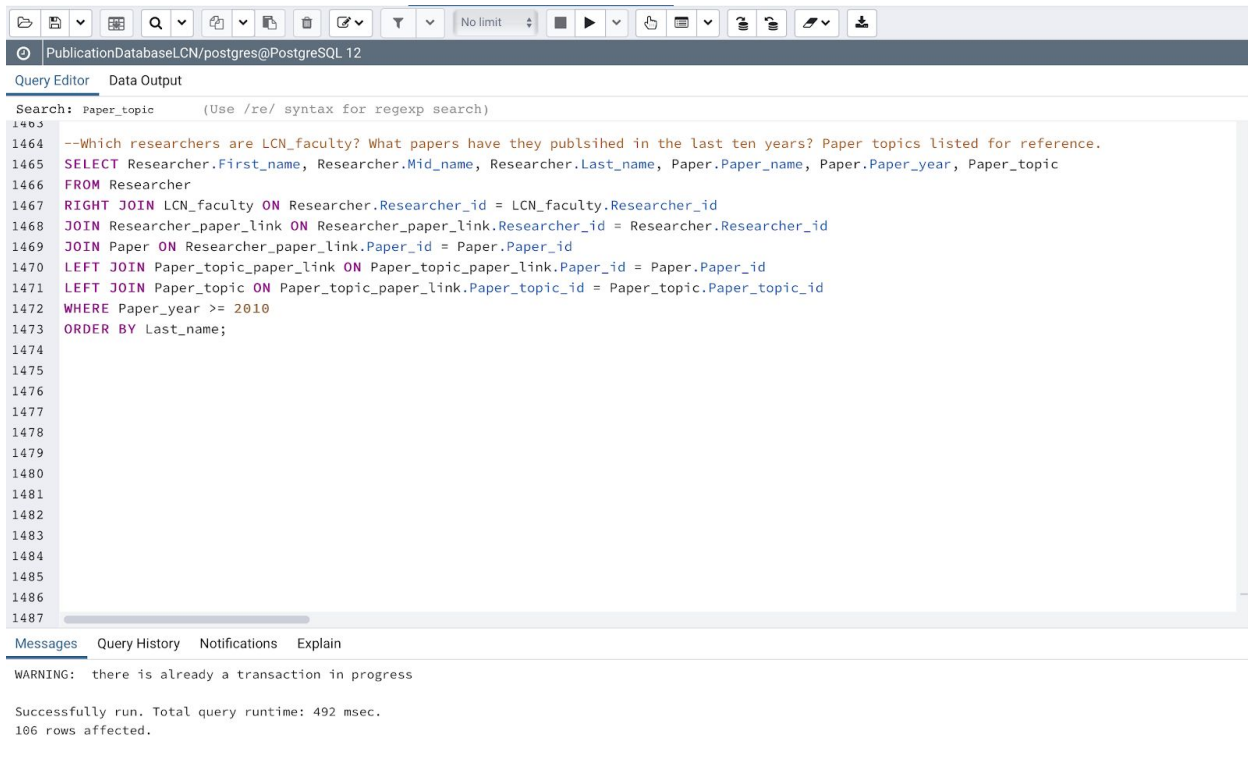
| citation_history_id [PK] numeric (12) | paper_id numeric (12) | old_citation_num numeric (6) | new_citation_num numeric (6) | change_date date |
|--|--------------------------|---------------------------------|---------------------------------|---------------------|
| 1 | 40001 | 20001 | 31 | 46 2020-06-23 |

PublicationDatabaseLCN Questions and Queries

Query 1

This query shows a useful question the database can be used to answer: Which researchers are LCN_faculty? What papers and their associated paper topics have they published in the last ten years?

This is a useful question because end users of the website application will want to familiarize themselves with the research of the faculty in the lab. Oftentimes potential collaborators or graduate students will want to read the papers that the researchers have recently published. Generally persons consider the papers published in the last ten years to be the most relevant to what the researcher is currently focused on. This is also a useful question for potential promotions, since recent publications are considered when faculty are promoted.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```

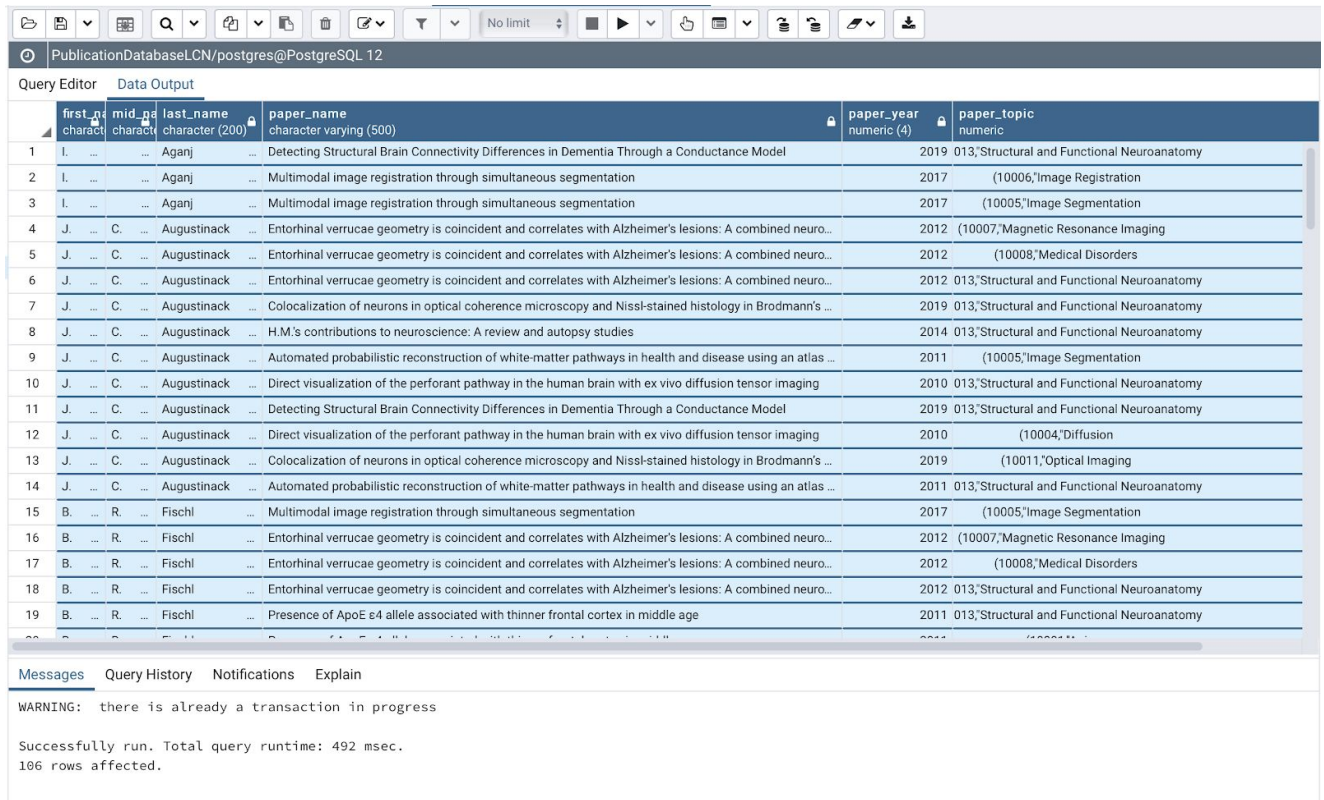
1464 --Which researchers are LCN_faculty? What papers have they published in the last ten years? Paper topics listed for reference.
1465 SELECT Researcher.First_name, Researcher.Mid_name, Researcher.Last_name, Paper.Paper_name, Paper.Paper_year, Paper_topic
1466 FROM Researcher
1467 RIGHT JOIN LCN_faculty ON Researcher.Researcher_id = LCN_faculty.Researcher_id
1468 JOIN Researcher_paper_link ON Researcher_paper_link.Researcher_id = Researcher.Researcher_id
1469 JOIN Paper ON Researcher_paper_link.Paper_id = Paper.Paper_id
1470 LEFT JOIN Paper_topic_paper_link ON Paper_topic_paper_link.Paper_id = Paper.Paper_id
1471 LEFT JOIN Paper_topic ON Paper_topic_paper_link.Paper_topic_id = Paper_topic.Paper_topic_id
1472 WHERE Paper_year >= 2010
1473 ORDER BY Last_name;

```

Below the query editor, the status bar shows: "WARNING: there is already a transaction in progress", "Successfully run. Total query runtime: 492 msec.", and "106 rows affected."

To obtain the answer to this question I started by selecting from the Researcher table since determining LCN_faculty is the first part of my question. To filter out researchers who are not LCN_faculty I did a RIGHT JOIN on the LCN_faculty subtype table, so that only relevant values from the rightmost table (LCN_faculty) remain. I then JOIN both Researcher_paper_link and Paper tables to obtain the papers that the LCN_faculty have published. I then did LEFT JOIN on Paper_topic_paper_link and Paper_topic to determine the topics of each paper. Lastly I used WHERE to keep only papers published within the last ten years, and ordered by LCN_faculty Last_name for readability and organization.

See the screenshot below for a small faction of the returned rows, since they were too many to include in one screenshot.



| | first_name | mid_name | last_name | paper_name | paper_year | paper_topic |
|----|------------|----------|-------------|--|------------|---|
| 1 | I. | ... | Aganj | Detecting Structural Brain Connectivity Differences in Dementia Through a Conductance Model | 2019 | 013,'Structural and Functional Neuroanatomy |
| 2 | I. | ... | Aganj | Multimodal image registration through simultaneous segmentation | 2017 | (10006,'Image Registration |
| 3 | I. | ... | Aganj | Multimodal image registration through simultaneous segmentation | 2017 | (10005,'Image Segmentation |
| 4 | J. | C. | Augustinack | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combined neuro... | 2012 | (10007,'Magnetic Resonance Imaging |
| 5 | J. | C. | Augustinack | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combined neuro... | 2012 | (10008,'Medical Disorders |
| 6 | J. | C. | Augustinack | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combined neuro... | 2012 | 013,'Structural and Functional Neuroanatomy |
| 7 | J. | C. | Augustinack | Colocalization of neurons in optical coherence microscopy and Nissl-stained histology in Brodmann's ... | 2019 | 013,'Structural and Functional Neuroanatomy |
| 8 | J. | C. | Augustinack | H.M.'s contributions to neuroscience: A review and autopsy studies | 2014 | 013,'Structural and Functional Neuroanatomy |
| 9 | J. | C. | Augustinack | Automated probabilistic reconstruction of white-matter pathways in health and disease using an atlas ... | 2011 | (10005,'Image Segmentation |
| 10 | J. | C. | Augustinack | Direct visualization of the perforant pathway in the human brain with ex vivo diffusion tensor imaging | 2010 | 013,'Structural and Functional Neuroanatomy |
| 11 | J. | C. | Augustinack | Detecting Structural Brain Connectivity Differences in Dementia Through a Conductance Model | 2019 | 013,'Structural and Functional Neuroanatomy |
| 12 | J. | C. | Augustinack | Direct visualization of the perforant pathway in the human brain with ex vivo diffusion tensor imaging | 2010 | (10004,'Diffusion |
| 13 | J. | C. | Augustinack | Colocalization of neurons in optical coherence microscopy and Nissl-stained histology in Brodmann's ... | 2019 | (10011,'Optical Imaging |
| 14 | J. | C. | Augustinack | Automated probabilistic reconstruction of white-matter pathways in health and disease using an atlas ... | 2011 | 013,'Structural and Functional Neuroanatomy |
| 15 | B. | R. | Fischl | Multimodal image registration through simultaneous segmentation | 2017 | (10005,'Image Segmentation |
| 16 | B. | R. | Fischl | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combined neuro... | 2012 | (10007,'Magnetic Resonance Imaging |
| 17 | B. | R. | Fischl | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combined neuro... | 2012 | (10008,'Medical Disorders |
| 18 | B. | R. | Fischl | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combined neuro... | 2012 | 013,'Structural and Functional Neuroanatomy |
| 19 | B. | R. | Fischl | Presence of ApoE ε4 allele associated with thinner frontal cortex in middle age | 2011 | 013,'Structural and Functional Neuroanatomy |

Messages Query History Notifications Explain

WARNING: there is already a transaction in progress

Successfully run. Total query runtime: 492 msec.
106 rows affected.

Query 2

This query shows a useful question the database can be used to answer: What are the top five journals with the highest impact? Which researchers are in these journals and what papers did they publish? Include the impact factor and journal topic for reference.

This is a useful question for potential promotions, since recent publications and the reput of the journals in which they are published are considered when faculty are promoted or considered for awards.

```

1476 --What are the top five journals with the highest impact? Which researchers are in these journals and what papers did they publish?
1477 --Impact factor and journal topic included for reference.
1478 SELECT Researcher.Last_name, Researcher.First_name, Paper.Paper_name, Journal.Journal_name, Journal_field_name, Journal.Impact_factor
1479 FROM Researcher
1480 JOIN Researcher_paper_link ON Researcher_paper_link.Researcher_id = Researcher.Researcher_id
1481 JOIN Paper ON Researcher_paper_link.Paper_id = Paper.Paper_id
1482 RIGHT JOIN Journal on Journal.Journal_id = Paper.Journal_id
1483 JOIN Journal_field ON Journal_field.Journal_field_id = Journal.Journal_field_id
1484 WHERE Journal.Journal_id IN (SELECT Journal.Journal_id
1485                             FROM Journal
1486                             JOIN Journal_field ON Journal_field.Journal_field_id = Journal.Journal_field_id
1487                             WHERE Journal.Impact_factor IS NOT NULL
1488                             ORDER BY Journal.Impact_factor DESC
1489                             LIMIT 5)
1490 GROUP BY researcher.last_name, Researcher.First_name, Paper.Paper_name, Journal.Journal_name, Journal_field_name,
1491          Journal.Impact_factor, Journal.Journal_id
1492 ORDER BY journal.impact_factor DESC;
1493
1494
1495
1496
1497
1498
1499

```

Messages Query History Notifications Explain

WARNING: there is already a transaction in progress

Successfully run. Total query runtime: 492 msec.
163 rows affected.

The joins done in this query are similar to the previous query. Of note in this query is the uncorrelated subquery in the WHERE statement. This subquery is used to limit the number of journals to the top five based on impact factor. First all journals are obtained using the SELECT FROM command, then the WHERE statement is used to get rid of null values since Postgres puts nulls above the highest value. The list is ordered in descending or using the ORDER BY command on the Impact factor values, then the top five are selected using the LIMIT command. This subquery fits into the outer query because the outer uses the WHERE command to compare all the returned journal IDs from the query to determine whether it is in the list of returned journals from the subquery. The outer query will then only return papers from journals that are in the list generated by the subquery.

See the screenshot below for a small faction of the returned rows, since they were too many to include in one screenshot.

| PublicationDatabaseLCN/postgres@PostgreSQL 12 | | | | | | |
|---|------------------------------|-------------------------|---|---|---------------------------------------|--------------------------------|
| Query Editor Data Output | | | | | | |
| | last_name character (200) | first_name character | paper_name character varying (500) | journal_name character varying (200) | journal_field_name character (100) | impact_factor numeric (5,3) |
| 1 | Hess | C. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 2 | Rabinovici | G. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 3 | Rademakers | R. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 4 | Rosen | B. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 5 | Fan | C. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 6 | Ferrari | R. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 7 | Wang | Y. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 8 | Yokoyama | J. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 9 | Boxer | A. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 10 | Wang | R. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 11 | Fischl | B. | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combine... | Acta Neuropathologica | Neuroscience | 18.174 |
| 12 | Wang | R. | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combine... | Acta Neuropathologica | Neuroscience | 18.174 |
| 13 | Fischl | B. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 14 | Wang | L. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 15 | Frosch | M. | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combine... | Acta Neuropathologica | Neuroscience | 18.174 |
| 16 | Wang | L. | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combine... | Acta Neuropathologica | Neuroscience | 18.174 |
| 17 | Wang | H. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 18 | Augustinack | J. | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combine... | Acta Neuropathologica | Neuroscience | 18.174 |
| 19 | Wald | L. | Entorhinal verrucae geometry is coincident and correlates with Alzheimer's lesions: A combine... | Acta Neuropathologica | Neuroscience | 18.174 |
| 20 | Bonham | L. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 21 | Hardy | J. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 22 | He | T. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |
| 23 | Dale | A. | Shared genetic risk between corticobasal degeneration, progressive supranuclear palsy, and fro... | Acta Neuropathologica | Neuroscience | 18.174 |

Messages Query History Notifications Explain

WARNING: there is already a transaction in progress

Successfully run. Total query runtime: 492 msec.
163 rows affected.

Query 3

This query shows a useful question the database can be used to answer: What journals have lost impact? Listing only journals with a new impact factor of less than two.

This is a useful question for potential promotions, since recent publications and the reput of the journals in which they are published are considered when faculty are promoted or considered for awards. For certain institutions an impact factor of 2 or below can be deemed as low, so papers published in those journals will have less reput than those published in higher tier journals. The change of the impact factor of a journal can be very relevant if a faculty is about to be promoted. The returned journal in this instance, "Annals of Neurology" moved form an impact factor of 10.244 to 1.765, meaning that it has lost a significant amount of its impact. This is very relevant information to researchers as they may stop submitting papers to this journal and favour higher impact journals.

The screenshot shows a PostgreSQL Query Editor window titled 'PublicationDatabaseLCN/postgres@PostgreSQL 12'. The query editor contains the following SQL code:

```

1608
1609
1610 --What journals have lost impact? Listing only journals with new impact less than 2.
1611 SELECT Journal.Journal_name, old_impact_factor, new_impact_factor, change_date
1612 FROM Journal_impact_factor_history
1613 JOIN Journal ON Journal.Journal_id = Journal_impact_factor_history.Journal_id
1614 WHERE new_impact_factor < old_impact_factor
1615 GROUP BY Journal.Journal_name, old_impact_factor, new_impact_factor, change_date
1616 HAVING new_impact_factor < 2;
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630

```

Below the query editor, the 'Data Output' tab is active, showing a table with the following columns: journal_name, old_impact_factor, new_impact_factor, and change_date. The table contains one row of data:

| journal_name | old_impact_factor | new_impact_factor | change_date |
|---------------------|-------------------|-------------------|-------------|
| Annals of Neurology | 10.244 | 1.765 | 2020-06-23 |

This query uses a JOIN on the Journal table to obtain the journal_name. The WHERE command is used to indicate that only journals where the new impact factor is lower than the old impact factor should be returned. The HAVING command is used to further limit the returned rows to only journals where the new impact factor is less than 2.

Summary and Reflection

The database I designed will contain the publication information for three PIs in my lab. Since they are on several publications together there will be no redundancies in publications in the database. The data from the database will be used to populate each PI's personal website with only the publications where they are an author. The website will be interactive to allow the viewer to sort based on number of citations, publication year, first author last name. The user will also be able to search for publications based on name and the journal in which it was published and the research field of the journal. Papers can be sorted into various topics of research that LCN is involved in using the attributes under the paper entity: Aging, Cortical modelling, Developmental, Diffusion, Image segmentation, Image registration, Magnetic resonance imaging, Medical disorders, Memory, Motion artifacts, Optical imaging, Positron Emission Tomography.

My aspirational goals for this database is for it to also store the PDF for each article. The PDFs from journals that are open access can be viewed by all users, but PDFs from journals that have a paywall will only be accessible to LCN lab members. That way the database is useful not just for non lab members who are curious about the PIs in the lab, but also for lab members who would like to access a paper. I am currently thinking that I can either ask partners to allow Partners login credentials to be accessible on the website, that way lab members do not need to

create a new account to access the PDFs. If this is not possible then lab members will need to create an account and this account information will be added to the database. I have accounted for this in my EERD by adding the specialization subtypes of Paywall_paper and Opensource_paper. There is a M:N relationship between Paywall_paper and a user Account.

It was exciting to add data to my database, run queries and see the actual operation of the database that I spent so long creating! I was able to finally have a concrete understanding of how each table relates to another. The database is normalized and there are key driven indexes and query driven indexes that would be commonly used for subqueries. The physical EERD has proven to be very useful throughout the process of creating the database. Both the conceptual and physical EERD have been updated to reflect the historical tables created. It's been a lot of work but it has been very rewarding to have actually created something at the end of a class. I can't wait to implement this on the website and see what the faculty think about the database. I'm starting to really get the idea of the "Applied" in the program name Applied Data Analytics. Now I can say I not only took the class but I actually know how to make a database.

Thanks so much for being an amazing facilitator Galina!

For questions:

Can I share access to my database with others on pgAdmin so they run queries like I can?

My Paper_topic in Query 1 came out kind of funny and I don't know why.