THE CYBERFEMINISM INDEX : AN SQL DATABASE

FOR CODE FIRST GIRLS

DATABASE OVERVIEW

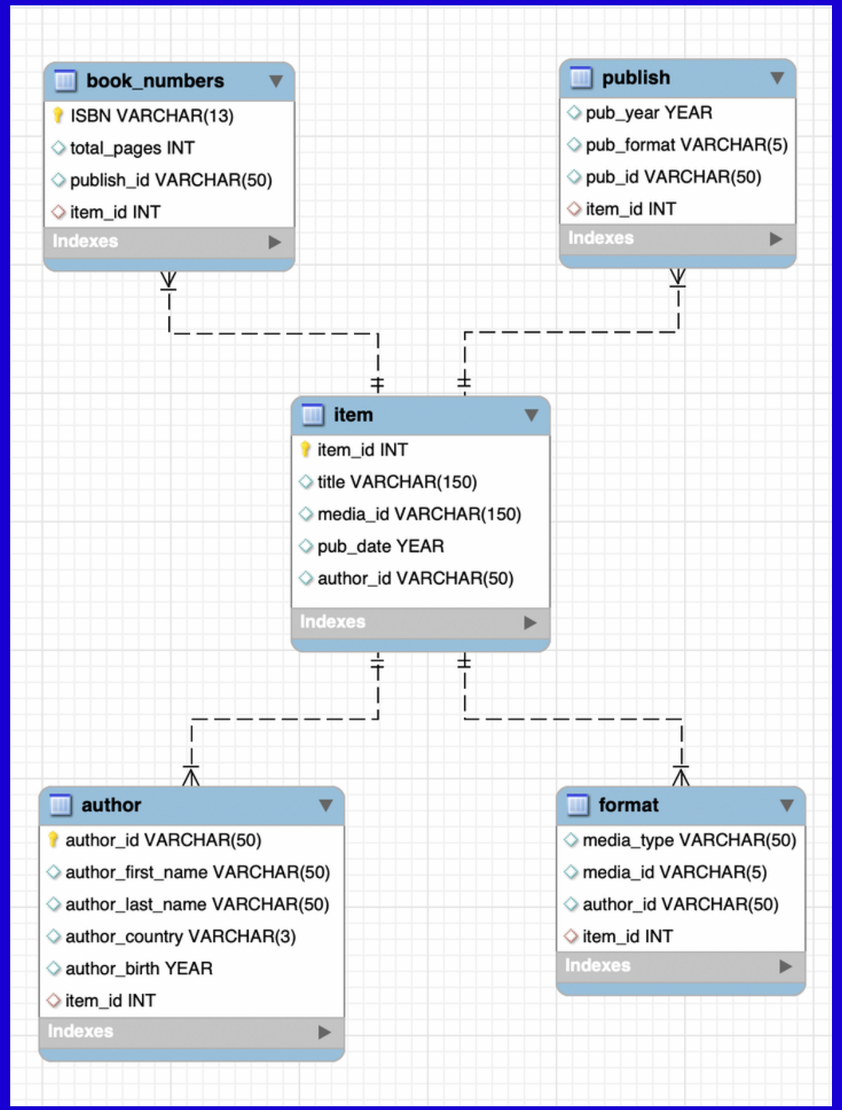
1. The Cyberfeminism Index, an ongoing archival project by Dr Mindy Seu, gathers three decades of online activism, academic research, literature and net art. Hackers, scholars, artists and activists worldwide have contributed to the database; submissions are ongoing.

2. Purpose : To act as an archive analyst for the project, to have a comprehensive set of information of submissions and their authors.

3. Aim : To ascertain at what time periods certain publication formats were popular, the number of authors from each country, as well as database insights, such as the years where the topic was more popular

DATABASE MAP

THE RELATIONSHIPS BETWEEN THE FIVE TABLES



JOINS

TYPE OF JOIN : RIGHT JOIN

Query Using Join : What are the author details for authors of published books?

This insight can be used to determine who authored the items in the dataset which are published books, and conversely, any authors of other types of media (e.g. journal articles, art, hypertext etc)

```
/*==== JOINING TABLES ===*/
#What are the author details for
#authors of published books?
USE cyberfeminism;

SELECT * From author;
SELECT * From book_numbers;

-- RIGHT JOIN--
SELECT A.author_id, A.author_first_name,
A.item_id, A.author_last_name,
A.author_country,A.author_birth,
B.ISBN,B.total_pages,B.pub_id
FROM Author AS A
RIGHT JOIN
Book_Numbers AS B
ON
A.item_id = B.item_id;
```

28:151

xplain Display Info: Read + Eval c

STORED FUNCTION

```
USE cyberfeminism;

SELECT * FROM cyberfeminism.item;

DELIMITER //
CREATE PROCEDURE insertitem (IN item_id INT, IN title VARCHAR (150),
 IN media_id VARCHAR (150), IN pub_date YEAR, IN author_id VARCHAR(50))
BEGIN
    INSERT INTO Item(item_id,title,media_id,pub_date,author_id)
    VALUES (item_id,title,media_id,pub_date,author_id);
END//
DELIMITER ;

CALL insertitem(782,'Glitch Feminism: A Manifesto','BOOK',2020,'LR');
```

TYPE OF FUNCTION: PROCEDURE

To be applied to a query

In this scenario, a new author has submitted a piece of their work into the index. As item is the PRIMARY KEY table, it will need to be at least inserted there.

This procedure makes it quick and easy to insert a new item!

STORED FUNCTION

```
/*====================STORED FUNCTION=======================*/
SELECT
    author_id,
    author_first_name,
    author_last_name,
    CONCAT(author_first_name, ' ', author_last_name) full_name
FROM
    cyberfeminism.author
ORDER BY
    full_name;
```

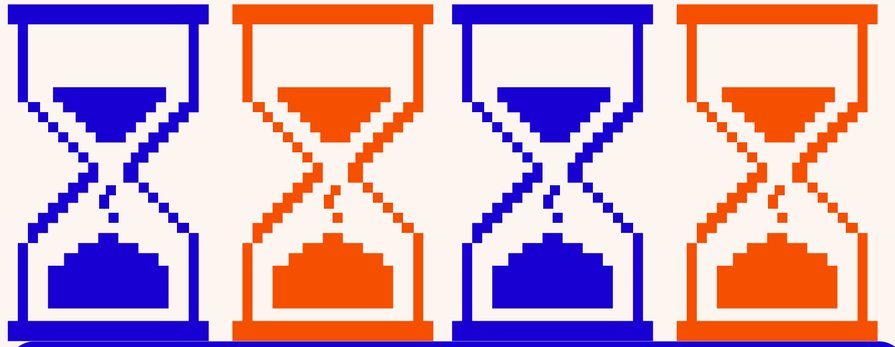
TYPE OF FUNCTION: CONCATENATION

To be applied to a query

In this scenario, we want to see the full names of authors, not just their first names or last names - we want a new item that provides their full name.

This procedure makes it quick and easy to see author's full names, which can make searching them up easier e.g. if we want to email them or look up their other works

```
/*==================CREATE AN EVENT===========*/
CREATE TABLE update_reminder
(ID INT NOT NULL AUTO_INCREMENT,
LAST_UPDATE TIMESTAMP,
PRIMARY KEY (ID));
DELIMITER //
CREATE EVENT reminder
on schedule at now() + interval 1 day
do begin
    insert into update_reminder(last_update)
    values ('Check the index for any updates!');
END //
DELIMITER ;
SELECT*FROM update_reminder;
```



TYPE OF FUNCTION: EVENT

In this scenario, we know that authors are going to be submitting to the index on a daily basis, and therefore we need a reminder to check for new submissions, or any corrections to previous submissions.

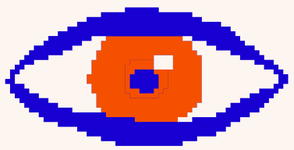
This event sends us a reminder every day to check the index, which ensures that the database is up to date!

SUBQUERIES

SUBQUERIES FOR DATA EXTRACTION

To be applied to a query

1. Finds out how many items were published each year. This is ideal for identifying patterns in the dataset (e.g. a particular type of item format, such as VR, was very popular from 1995-1997)
2. Finds out how many items were published by author's nationality. Seeing any skewness in this distribution could highlight a need for calls for submissions in less represented countries.
3. Provides an overview of media types that aren't published books; a more precise alternative than 'Books and Other'.



```sql
/*====== EXAMPLE QUERY WITH A SUBQUERY FOR DATA EXTRACTION ================*/
-- Find out how many items were published each year (in no particular order)
SELECT pub_date, COUNT(pub_date)
FROM item
GROUP BY pub_date;
-- What are the names of authors with published books?
SELECT author_first_name, author_last_name
FROM author
WHERE item_id IN (
    SELECT item_id
    FROM book_numbers
);
-- Find out how many items were published by author's nationality
SELECT author_country, COUNT(author_country)
FROM author
GROUP BY author_country;
-- What are the media types that aren't published books?
SELECT media_type,media_id
FROM Format
WHERE NOT EXISTS (
  SELECT *
  FROM book_numbers
  WHERE Format.item_id = book_numbers.item_id
);
```

WHAT I WOULD DO NEXT TIME

I would relate the Book_Numbers table directly to the Publish table, to ensure that the relationships between datasets were as parsimonious as possible

Create views of each item across multiple datasets to assess all the information we have about each submission

I would also add tables on other aspects - such as the qualities of the items (main theories, academic fields etc)

THANKS

HERE'S A LINK TO DATA SOURCE IF YOU WANT TO CHECK IT OUT:
HTTPS://CYBERFEMINISMINDEX.COM