WOLFPUB DB

For WolfCity publishing house

CSC 540 Database Management Systems
Project Report #1

Feb 17, 2020.

Swathi Dinakaran(sdinaka)
Sameer Thummalapally(vthumma)
Sri Harsha Varma Uppalapati(suppala)
Sai Rama Mohan Reddy Singamala(ssingam2)

Contents

1. Problem Statement.....3

2. Intended Classes of Users of The Database System.....3

3. Five Main “Things”.....4

4. Realistic Situations.....4

DATABASE System Assumptions.....4

5. The API’s.....5

6. Descriptions of The Views of The Data.....8

7. Local E/R Diagrams.....9

8. Description of the local E/R Diagrams.....12

9. Local Relational Schema.....13

10. Document Of The Local Relational Schemas.....14

1.Problem Statement:

We have to develop a WolfPub Database Management System for Wolf city publishing house which can be used by the editors and management of the publishing house. The database system should have information about the publications, editors, authors, distributors, and journalists. We must also be able to perform operations such as entering the basic information of the publication, editing it, entering the information of distributors and also maintaining and generating reports in terms of orders made by the distributors such as the book they ordered, the order price, payments made to the employers, revenue of the publishing house.

The reason why we feel a database is a good option over others is that the database system provides various advantages such as efficiency in terms of database storage fetching the information, easy to store and access the data, security to the information. Moreover using a database system would allow users to create their own database, specify the schema for it and even query the data using a programming interface. File Systems do not have a transaction manager, contrary to the file system, database management systems have a transaction manager and in the case of WolfPubDb transaction management is one of the key aspects that need to be implemented because of the fact that there are various users of the system who try to access and update the available data concurrently and the absence of a transaction manager would lead to a lot of problems such as incorrect or redundant data.

2.Intended Class of Users:

Management: Admins are the people responsible for maintaining all the information related to the Database. They have the right to perform all the possible operations such as entering, editing, deleting the information of the staff, publications, distributors.

Editors: Editors can see the publications for which they are responsible. They can also edit, delete them. They have certain payment records.

Authors: Authors write books and are responsible for the books they have written. They have certain payment records.

Journalists: Just like authors, journalists are responsible for the articles they have written. They have certain payment records.

Distributors: Distributors place orders to the publication house. They make the order for a certain publication in a single instance.

3.Five Main Things:

1. Staff information: This entity would hold all the information related to the employers of the publishing house such as authors, editors, journalists.
2. Publications: This entity would hold all the information related to publications such as their name, topic, price.
3. Distributors: This entity holds the information related to distributors like their account id, name, city, how much they owe the publication house.
4. Payment History: This entity holds information regarding salary payments respective to the staff.
5. Orders: Orders is a relationship between distributors and the publications and determines information specific to the orders made by the distributors such as the publication they order, quantity and amount for the order.

4.Realistic Situations:

Situation-1: Imagine that Swathi is an editor of the publication house and she would like to review the articles she wrote. When she is viewing one of them, she gets to know that she made an error in it and she edits the publication.

Situation-2: Imagine Harsha is an investigative journalist and while writing an article he wanted to refer to an old article that is similar to this. So he asks the admin to provide him that article and gives him details of the article like the author, topic, and date it was published. Then the author searches in the DB to find that specific article.

Assumptions:

1. Each article has exactly one journalist.
2. Distributors can order only one type of publication at a time.
3. Every Publication will have a unique id, which shall be created at time of insertion.
4. Editors, authors will be paid as salary for staff employees and pay for invited employees.
5. Journalists are a part of the staff, and are paid just as editors and authors.
6. Staff members have a unique employee id.
7. Payments shall be done at the end of every month.
8. A book can have one or more authors.

5.Application Programming interfaces:

- *Editing and publishing:*
    - *enter_publication(title ,topic, price, type)*
        Return API for enter_book or enter_article based on type

    - *edit_publication(title ,topic, price, type)*
        Return API for edit_book or edit_article based on type

    - *enter_book(edition,title,topic, price, ISBN,date_of_publication, authors)*
        Return option as API to enter_chapter
            Or confirmation
            - *Can't update if ISBN,edition, title is null*

    - *edit_book(edition,title,topic, price, ISBN,date_of_publication, authors)*
        Return option as API to enter_chapter
            Or confirmation
            - *Can't update if ISBN,edition, title is null*

    - *Enter_periodic issue(title,topic, price,date_of_issue, periodicity)*
        Can't update
        Return confirmation

    - *Edit_periodic issue(title,topic, price,date_of_issue, periodicity)*
        Can't update
        Return confirmation

    - *assign_editors(staff.id, publication.id)*
      Return confirmation
      Can't update if any value is null

    - *Edit_table of contents(operation type):*
      Return api based on the operation type

    - *editor_view(staff.id,):*
      Return records of publication as per the staff id

- *Production of a book edition or of an issue of a publication.*
    - *enter_chapter(title, text):*
      Return confirmation

    - *edit_chapter(title, text):*
      Return confirmation

    - *Enter_article(publication_id, title, date_of creation)*
      Return confirmation

    - *Edit_article(publication_id, title, date_of creation,)*
      Return confirmation

    - *delete_book(ISBN):*
      Return confirmation or error if ISBN does not exist

    - *delete_periodic_issue(title. Topic, price, date_of_issue, periodicity):*
      Return confirmation

    - *find_book(topic,date_of_publication, author)*
      Return book or books as records, based on the params given

    - *find_article(title, date_of_creation, journalist):*
      Return article or articles as records, based on the params given

    - *enter_payment(staff.id, amount, timestamp):*
      Return confirmation
      If any value is null cant be updated

- *Distribution. Input orders from distributors, for a book edition or an issue of a publication per distributor, for a certain date.*
    - *add_distributer(name, type, street address, city, phone number,contact_person, accout_number)*
      Return confirmation.If the account number name, contact person is null, it can't be updated.

    - *edit_distributer(name, type, street address, city, phone number,contact_person, accout_number)*
      Return confirmation.If the account number name, contact person is null, it can't be updated

- *delete_distributer(name, type, street address, city, phone number,contact_person, accout_number)*
  Return confirmation.If the account number name, contact person is null, it can't be deleted

- *bill_distributer(owe, account_number):*
  Return confirmation after updating owe attribute for the distributer identified by the account number given

- *change_balance(amount_recieved, account_number):*
  Return confirmation after updating the amount owed by subtracting the amount_recieved.

- *add_order(number_of_copies, edition/issue, date_of_order, date_to_be_delivered, shipping_cost,total_order_value)*

- *Reports*

  - *generate_report(from_date, to_date):*
    Return price and copies of publication aggregated by distributer per month(given as from_date and to_date)

  - *get_total_revenue():*
    Return total income for all distributors

  - *total_revenue_per_city_distributor_location():*
    Return revenue as records aggregated by city, distributor, and location.

  - *get_total_expenses():*
    Return total_expenses from payment_history and shipping cost from orders relation

  - *get_total_distributor_number():*
    Return total number of distributors

  - *Calculate_payments_to_editors_and_authors(from_date, to_date, work_type):*
    Return total payments as records to the editors and authors, aggregated per time period and per work type.

6.Description of Views:

○ **Administrators View** - will have access to everything in the database. She/He can view all Staff Information, look at all the staff members, payment history, salaries of employees, list of publications and also how much each distributor owes to the publication house.

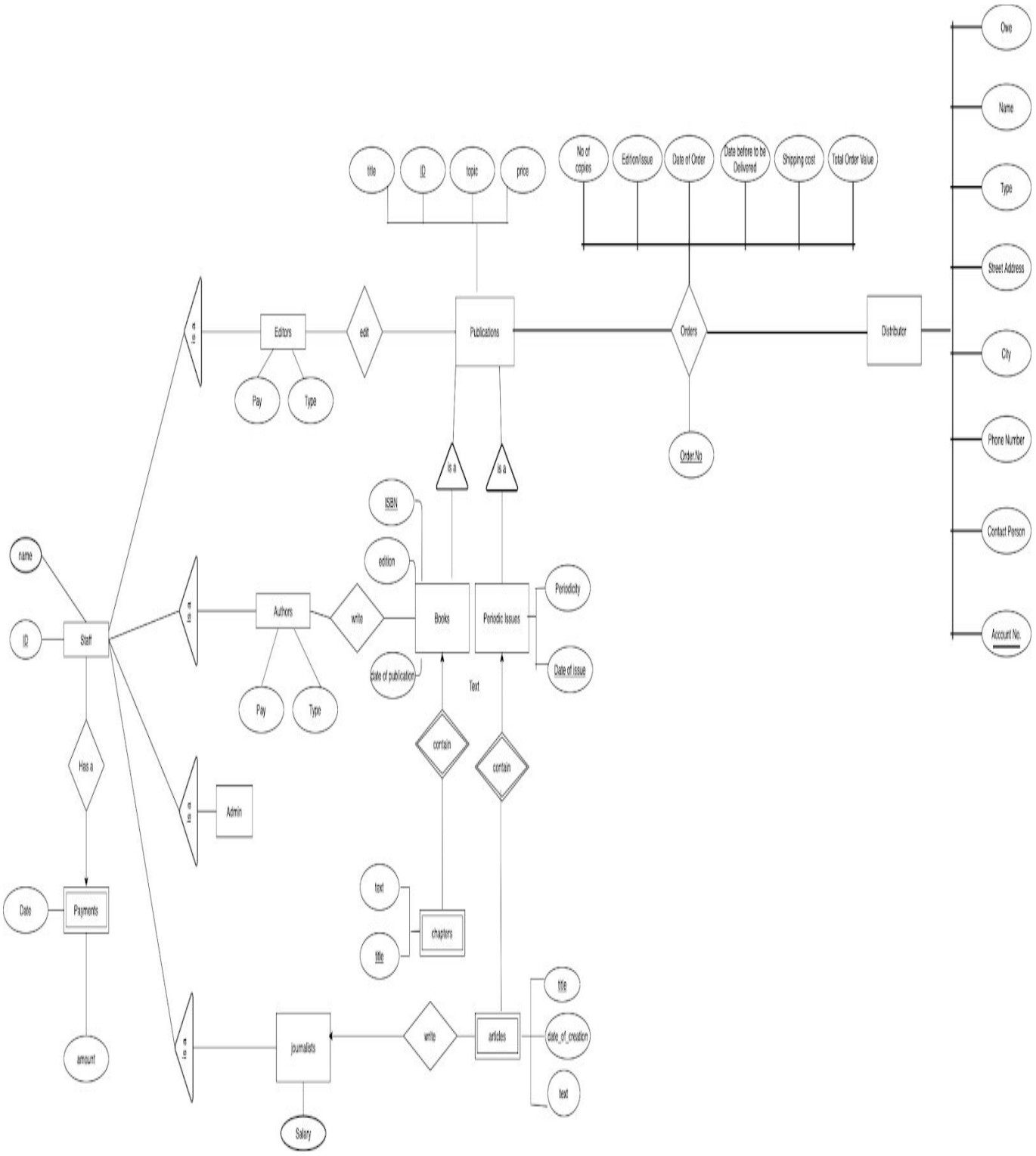
○ **Editor's View** – will have access to all the publications, Books and articles where he can check what are the publications assigned to him and also edit the contents of the book or articles of periodic issues. Editor or any other employee will not have access to salary information of other employees. He will also have access to check only his salary and payment history

○ **Author's View** - will have access to all the Books and its chapters. Author will be able to add chapters to the book or add books to the database based on the need. He will also have access to check only his salary and payment history.

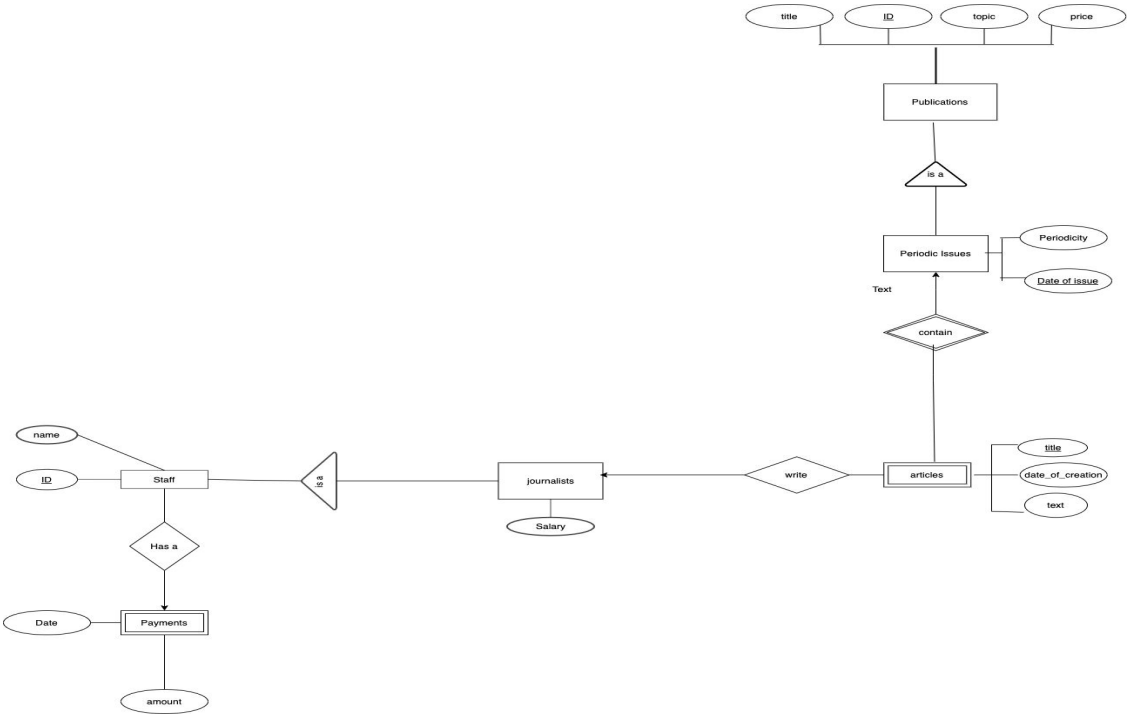
○ **Distributor's View** - Distributor will have access to all the publications of the WolfCity Publishing house, so that he can order the books or periodic issues he would like to. He would not access to edit the chapter or articles of the books/Issues as he is not authorized to access them.

○ **Journalists View** - will have access to all the Articles and its text. Author will be able to add text to articles or add articles to the database based on the need before publication of the issue. He will also have access to check only his salary and payment history.

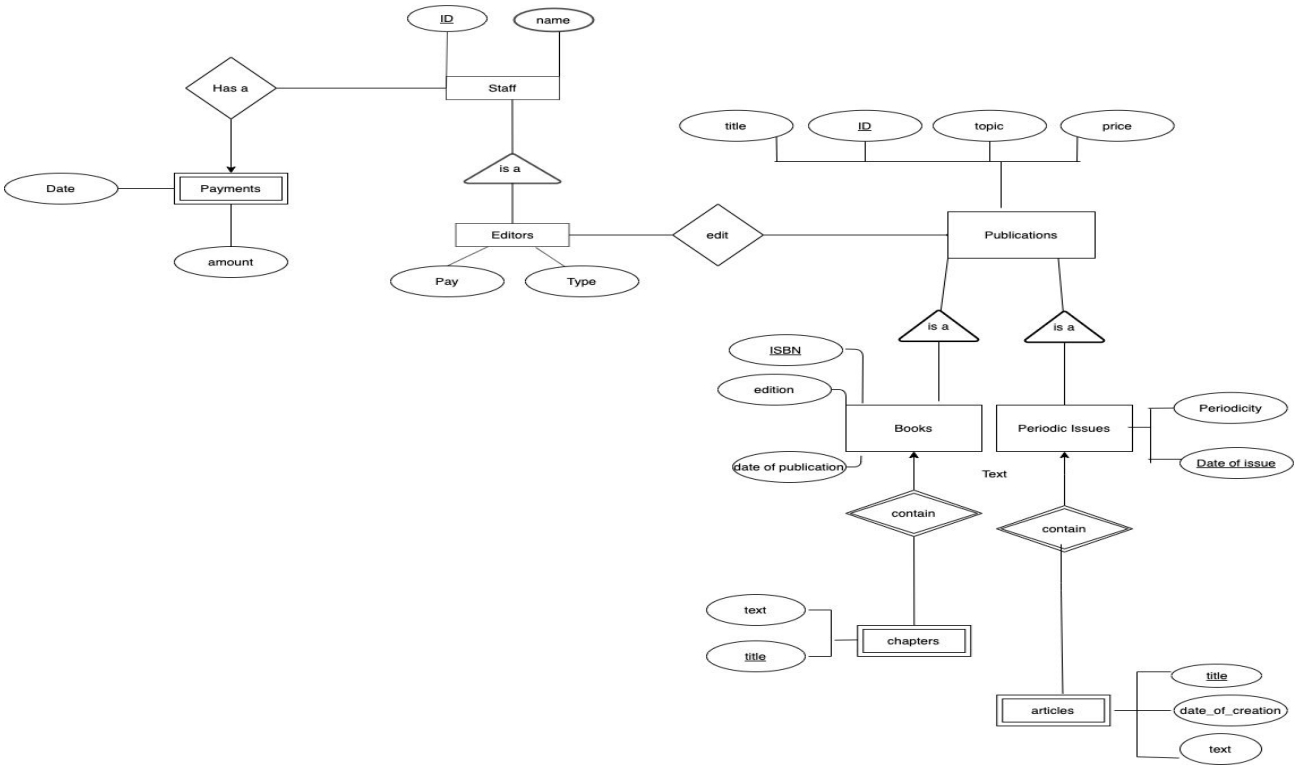
7.Local E/R Diagrams



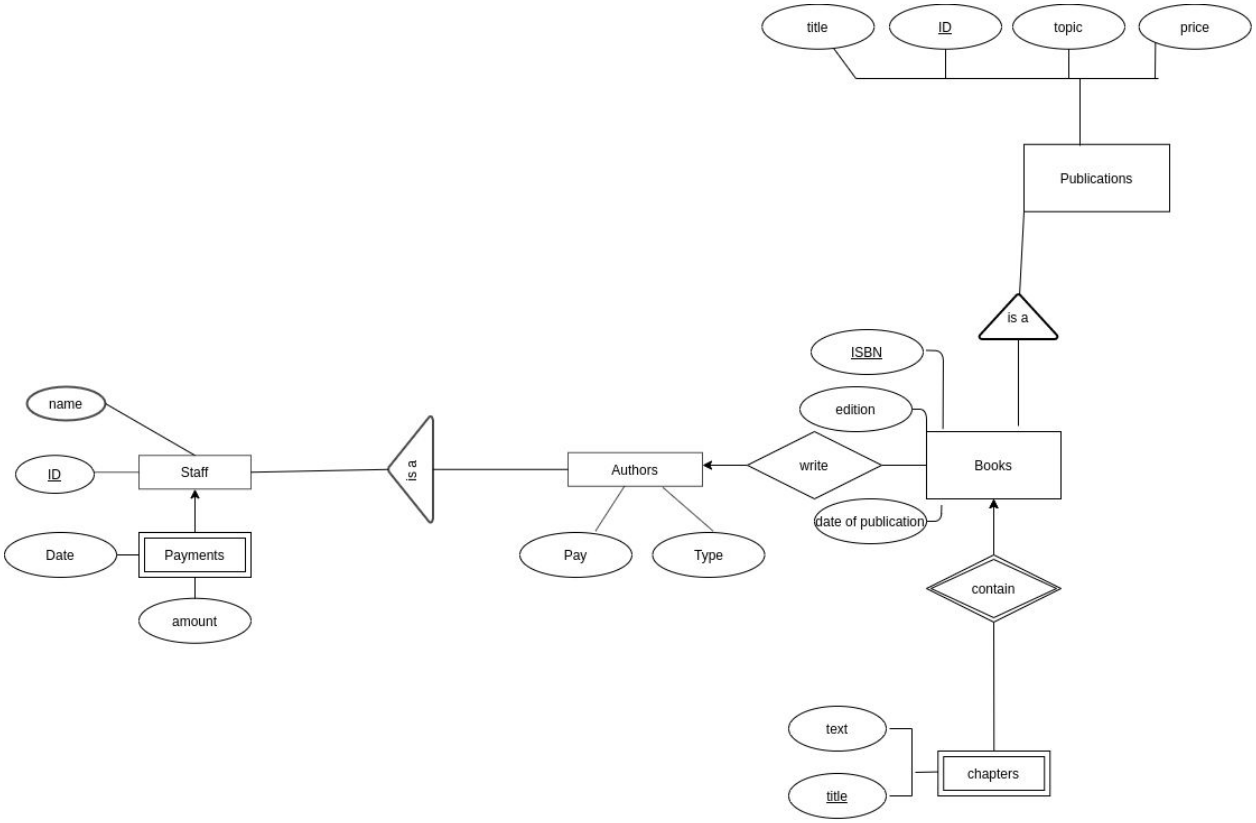
MANAGEMENT VIEW



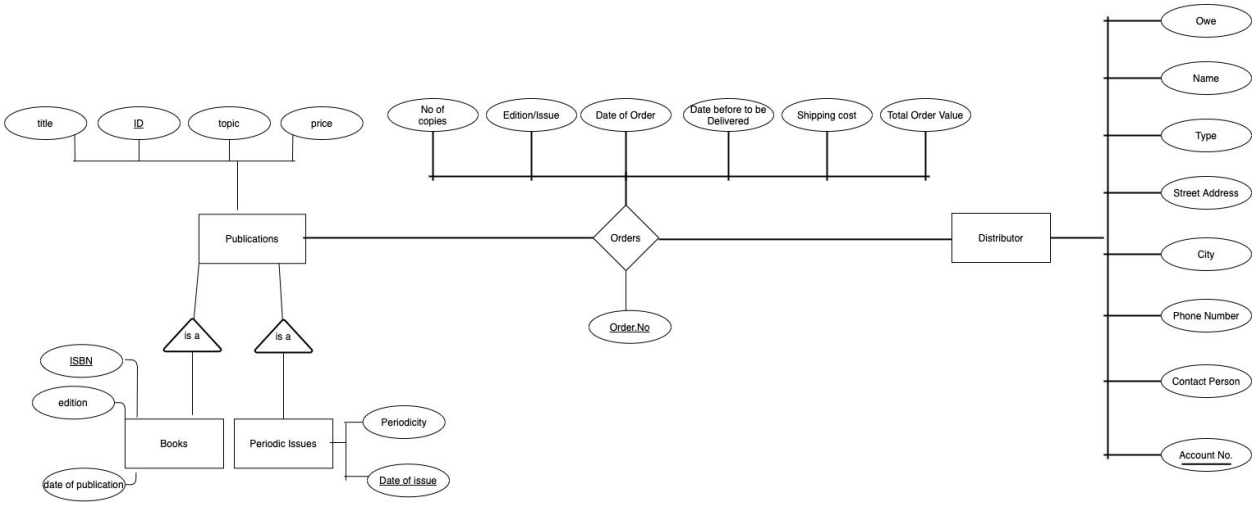
JOURNALIST VIEW



EDITORS VIEW



AUTHORS VIEW



DISTRIBUTOR'S VIEW

8. Description of Local E/R diagrams:

■ Publications is the main entity and it can publish Books or Periodic issues which are considered to be the subclasses of publications.
■ Staff is the main entity and it consists of Editors, Authors and Journalists.
■ Publications use unique Publishing ID created (mentioned in the assumptions) for every book or periodic issue and is taken as the primary key attribute.
■ Distributors have account ID as the primary key which is unique for every distributor.
■ Employee id has been created (mentioned in the assumptions) for every employee and is used as primary key as they don't have any unique identity.
■ Chapters is a weak entity set that uses ISBN and chapter title as the key, since chapters cannot exist without a book.
■ Articles is also a weak entity set that uses publishing id and article name as the key, since articles cannot exist without an issue.
■ Payments use employee id and date as the key, since payments cannot exist without an employee.
■ Orders is a many to many relation as a distributor can place multiple orders and the publishing house also has so many distributors.
■ Each periodic issue can have multiple articles, but each article can only be a part of at most one periodic issue.
■ Each book can have multiple chapters, but each chapter can only be a part of at most one book as specified by the narrative.
■ Each article is written only by one journalist and one journalist can write many articles. So, written by will be many to one relation from articles to Journalist.
■ Books can have multiple authors and each author can write multiple books which makes "written by" relation to be many to many relation.
■ Each Publication can have multiple editors and each editor can edit multiple publications which makes "Edit" to be many to many relations.
■ Payments contain payments related to multiple staff but one payment can belong to only one employee which makes it many to one relation from payments to employee.

9. Local Relational Schemas:

Distributors(Account Number, Name, Type, Street Address,City, Phone number, Contact Person)

Orders(Order number, Number of copies, edition/issue, Date of order, Date to delivered by, Shipping cost, Total Cost)

Publications(Publication ID, Topic, Title, Price)

Books(Publication ID, ISBN Number, Edition, Date of publication)

Chapters(Publication ID, ISBN Number, Title, Text)

Periodic Issue(Publication ID, Date of issue, Periodicity)

Articles (Publication ID, Date of issue,Title, Date of creation, Text)

Staff (Staff ID,name)

Editors(Staff ID, Type, Pay)

Authors(Staff ID, Type, Pay)

Journalists(Staff ID, Salary)

Payment History(Staff ID ,Transaction number,Date, Amount)

10. Local Schema Documentation:

Distributors(Account Number, Name, Type, Street Address,City, Phone number, Contact Person)

All the attributes of Distributors entity set are made to Distributor schema's attributes. However, it does not need to bring attributes from other entity sets because Distribution entity set is not a weak entity in our ER Diagram.

Orders(Order number, Number of copies, edition/issue, Date of order, Date to delivered by, Shipping cost, Total Cost)

The Orders relation is in between Distributor entity set and publication entity set.

Publications(Publication ID, Topic, Title, Price)

All the attributes of Publications entity set are made to Publication schema's attributes. However, it does not need to bring attributes from other entity sets because Publication entity set is not a weak entity in our ER Diagram.

Books(Publication ID, ISBN Number, Edition, Date of publication)

The attribute Publication ID is inherited from the entity set **Publications** as there exists a **is a** relation.

Chapters(Publication ID, ISBN Number, Title, Text)

Chapters is a weak entity as it does not exist without Books.The attribute Publication ID and ISBN Number are inherited from the entity set **Books** entity set**.**

Periodic Issue(Publication ID, Date of issue, Periodicity)

The attribute Publication ID is inherited from the entity set **Publications** as there exists a **is a** relation.

Articles (Publication ID, Date of issue,Title, Date of creation, Text)

Articles is a weak entity as it does not exist without Periodic issues.The attribute Publication ID and Date of issue are inherited from the entity set **Periodic Issues** entity set**.**

Staff (Staff ID,name)

All the attributes of a Staff entity set are made to Staff schema's attributes. However, it does not need to bring attributes from other entity sets because the Staff entity set is not a weak entity in our ER Diagram.

Editors(Staff ID ,type, pay)

The attribute Staff ID is inherited from the entity set **Staff** as there exists a **is a** relation.

Authors(Staff ID, type, pay)

The attribute Staff ID is inherited from the entity set **Staff** as there exists a **is a** relation.

Journalists(ID, Salary)

The attribute Staff ID is inherited from the entity set **Staff** as there exists a **is a** relation.

Payment History(Staff ID , Date,Amount)

Payment History is a weak entity as it does not exist without Staff ID.The attribute Staff ID is inherited from the entity set **Staff.**

Combining Many-One Relationships:

1) The relation between Payments and Staff is a many to one relation, as each staff can have multiple payments in their history.
2) The relation between Chapters and Book is a many to one relation, as each Book can have multiple chapters in their history.
3) The relation between Articles and Periodic issues is a many to one relation, as each Periodic issue can have multiple Articles in their history.