WolfPubDb

For the WolfCity publishing house

CSC 540 Database Management Systems
Project Report 1

Team Members

Suraj Sunil Pawar (spawar2)
Rishal Kamlesh Shah (rshah27)
Ankit Arvind Tiwari (atiwari4)
Amit Prafullkumar Ghetiya (aghetiy)

February 16th, 2020

Assumptions:

1. Articles were written at any point in time before publication released an issue, so they are different entities.
2. Books have editions and articles have issues.
3. Articles can exist without an issue.
4. One Article can be published in multiple issues even the same article can be available in another issue.
5. One contributor works for one publication house.
6. The date of payment to a contributor is the same as the date when the contributor claims the payment.

1. Problem Statement:

WolfPubDb is a database management system for the WolfCity publishing house which will store publication details of books, journals, and magazines, along with the subsequent editions and issues. These articles and books are written by editors and authors who are paid for their work either periodically or on a one-time basis pertaining to their job type. The distributors that are wholesale distributors, bookstores and libraries, place orders to the publishing house for editions of books or the issues of the articles. This is all done by keeping an account between the publishing house and the distributor which also gives a reference of all the current and past orders between the two of them. Thus, the WolfPubDb should be able to hold and maintain all the information pertaining to the publication house and retrieve it when necessary.

As mentioned in the problem statement a system should be able to hold, maintain and retrieve data efficiently when multiple users are accessing the data simultaneously which is not possible in case of the file system. The use of databases allows CRUD operations and improves performance drastically.

2. Intended Users:

1. **Distributors:** Add and retrieve orders with additional details like publication house and order details.
2. **Publication house management staff**: Add, update details of the book and articles.
3. **Contributors**: They mostly consist of authors and editors. So, the database will store their payment details like payment cycle, salary, and last payment.

3. Five Main Entities:

1. Publication (id, title, type, periodicity, typical topics)
2. Contributors (id, salary, job type, designation, name, id, payment cycle)
3. Book (ISBN, author, date of creation, content, publication date)
4. Order (id, shopping cost, price, date, no of copies, package id, order id, payment status)
5. Article (id, content, author, date of creation)

4. Tasks and Operations- Realistic Situations:

- *Situation 1:* A new publication named “Delta Ray” wants to get registered into the database so the administrator will enter all necessary information along with their book, author and distributor details into the system.

- *Situation 2:* An author just found a discrepancy in his previous payments so he wants to get details about all the previous payments he had received so he can reach administrator who has access to all views of the database management system.

5. Application Program Interfaces:

Editing and Publishing

- addPublication(PublicationID, title, type, periodicity, typical topics)
  return confirmation

- updatePublicationInfo(PublicationID, title, type, periodicity, typical topics)
  return confirmation
  Null if no entry exists

- deletePublicationInfo(PublicationID)
  return confirmation
  Null if no entry exists

- addEditor(contributorID, publicationID)
  return confirmation

- viewPublicationInfo(contributorID, publicationID)
  return title, type, periodicity, typical topics

- addArticleToIssue(issueID , articleID, author, date_of_creation, content)
  return confirmation

- deleteArticleFromIssue(issueID, articleID)
  return confirmation
  Null if no entry exists

- addChaptersToBook(ISBN , content)
  return confirmation

- deleteChapterFromBook(ISBN, chapter_id)
  return confirmation
  Null if no entry exists

Production of Book / Issue

- addNewBookEdition(ISBN, edition_no, author, publication_date, creation_date, content)
  return confirmation

- addNewIssue(issueID, issue_date)
  return confirmation

- updateBookEdition(ISBN, edition_no, author, publication_date, creation_date, content)
  return confirmation
  Null if no entry exists

- updateIssue(issueID, issue_date)
  return confirmation
  Null if no entry exists

- enterArticleText(articleID, content)
  return confirmation

- getListOfBooks(author, creation_date, topic)
  return list of books(ISBN, edition_no, author, publication_date, creation_date, content)
  Null if no entry exists for any input parameters.

- getListOfArticles(author, creation_date, topic)
  return list of articles(id, author, creation_date, content)
  Null if no entry exists for any input parameters.

- enterContributorPayment(contributorID, publicationID, designation, amount, payment_date)
  return list of payment log(contributorID, publicationID, designation, amount, payment_date)
  Null if no entry exists for any input parameters.

Distribution

- addDistributor(distributorID, name, address, city, contact_person, balance, phone_number, type)
  return confirmation

- updateDistributorInfo(distributorID, name, address, city, contact_person, balance, phone_number, type)
  return confirmation

- deleteDistributor(distributorID)
  return confirmation

- inputOrder(orderID, distributorID, date)
  return confirmation

- billDistributor(distributorID, orderID, shipping_cost, price, payment_status)
  return confirmation // Change balance associated with the distributor

- updateBalance(distributorID, payment_amount)
  return confirmation

Reports

- getMonthlyReport(distributorID, start_date, end_date)
  return list of record(publicationID, number_of_copies, total_amount) whose date is between start_date and end_date // Aggregate all rows with same Publication id

- totalCopies(packageID, distributorID)
  return number of copies

- getTotalRevenue(publicationID)

  return sum of all order amount associated with each distributor

- getTotalExpenses(publicationID)

  return sum of salary payment for each contributor + sum of shipping cost associated with each order shipment

- getNumberOfDistributors(publicationID)

  return number of unique distributorID counted using distributor table

- getTotalRevenue(distributorID, city, address)

  return sum of the price of orders per city, distributor, and location

- getTotalPayment(time_period, work_type)

  return sum of payment amount made by publication

6. Description of Views:

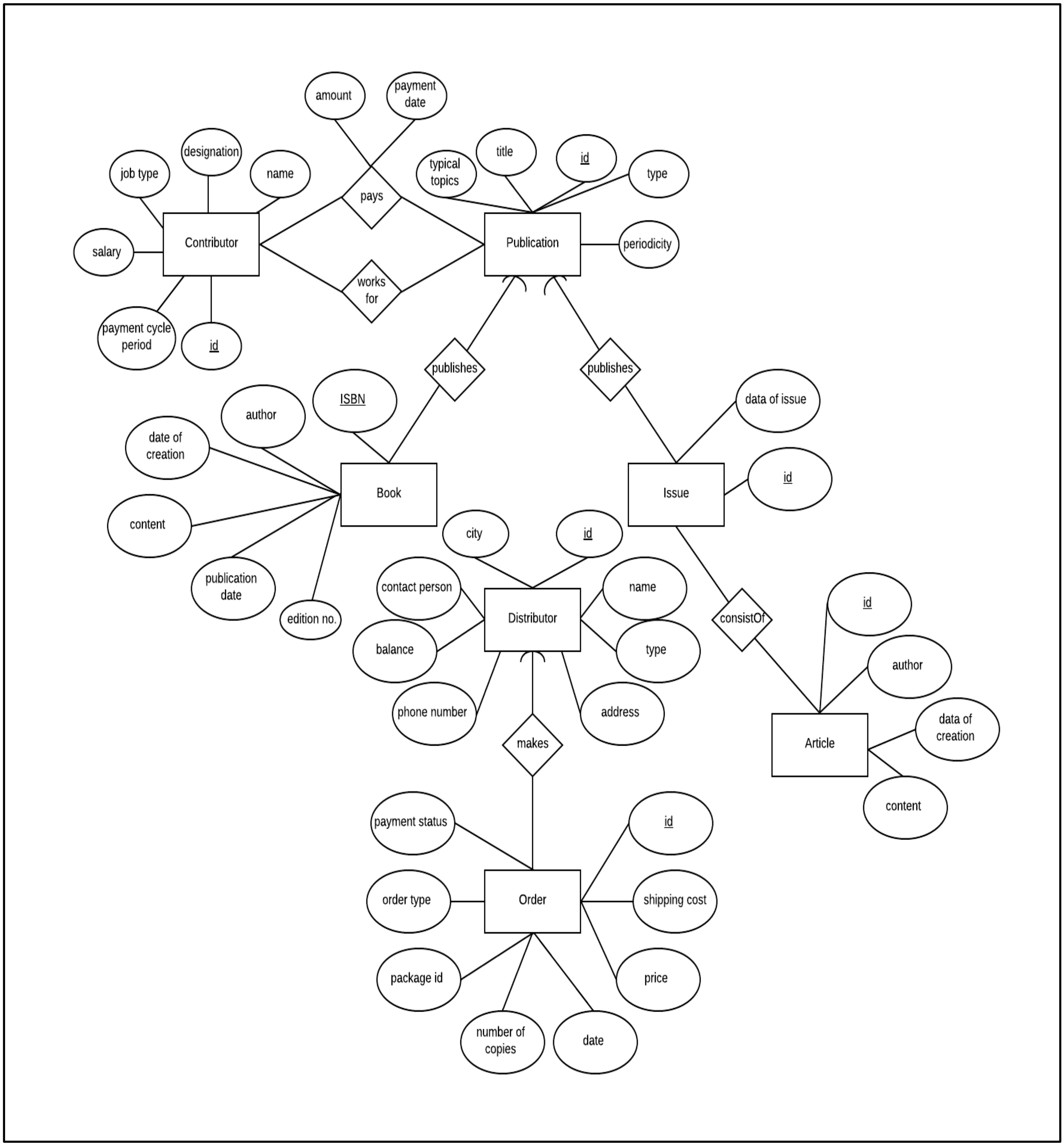
- **Administrator View:** This view will be for the administrative staff working at the Publishing House and will be accessible only to them. The view will provide visibility of the data containing the books, journals and magazines published and it will also contain the authors and editors linked and working for WolfCity.

- **Contributors:** This view is for the authors and editors writing the books and articles for the publishing house. The data in this view will pertain to the job type (staff or invited), their payment details like payment cycle, salary, and last payment date.

- **Distributors:** The distributors are the ones which order editions and issues from the publishing house. So the data related to the distributors will be visible in this view. The data will be details of the distributors themselves, their account information with the publishing house, and details related to their orders.

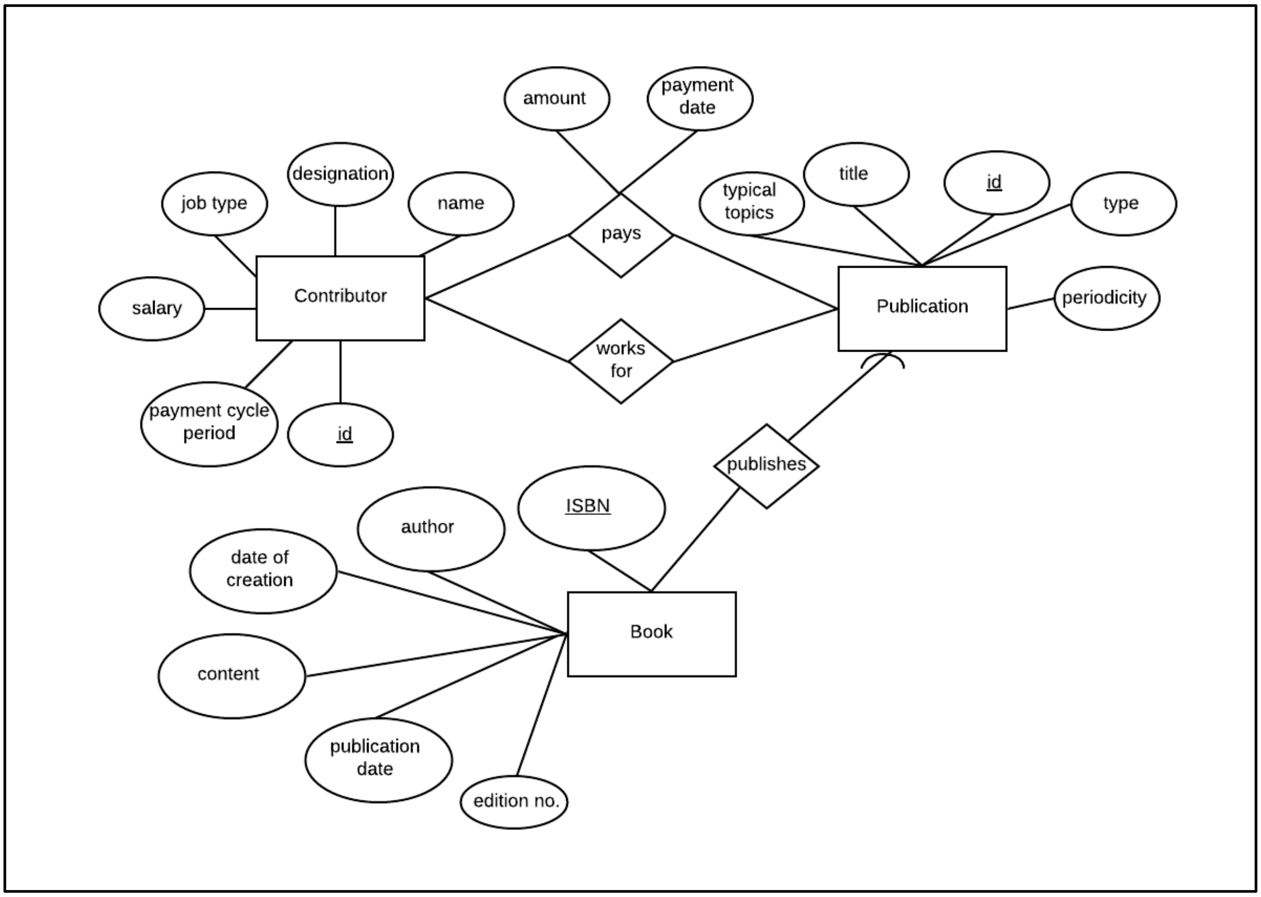
7. Local E/R Diagrams:

Admin's view:



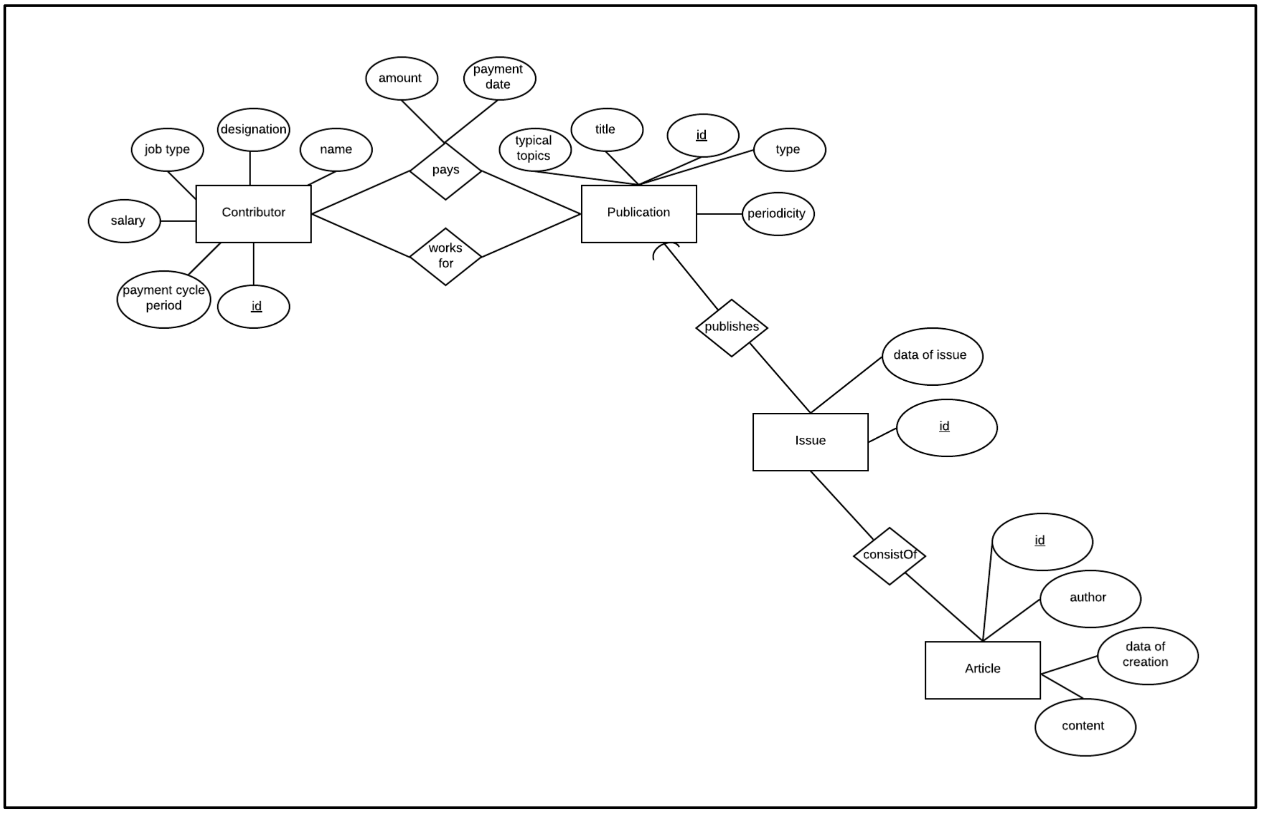
Admin's view

Author of a Book's view:



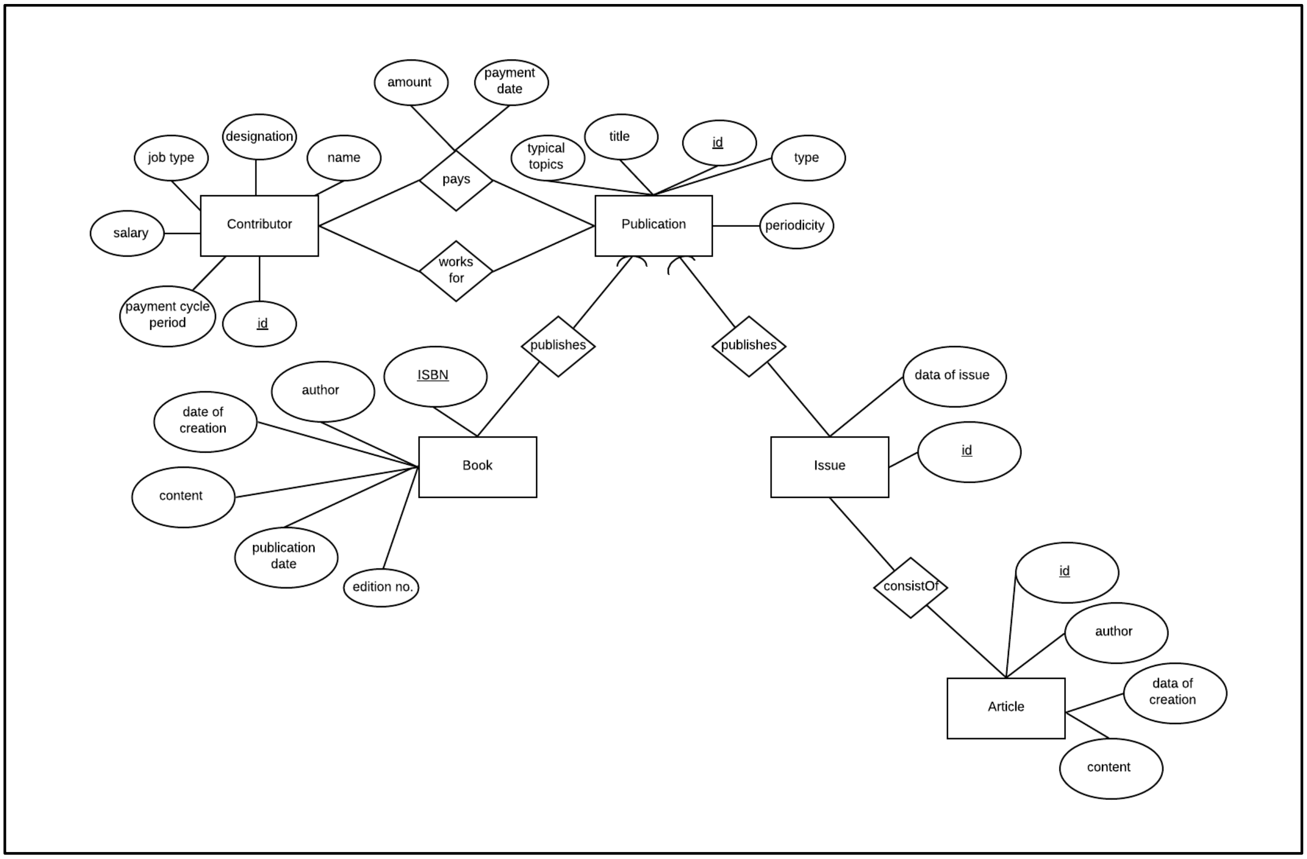
Book's view

Author of an Issue's view:



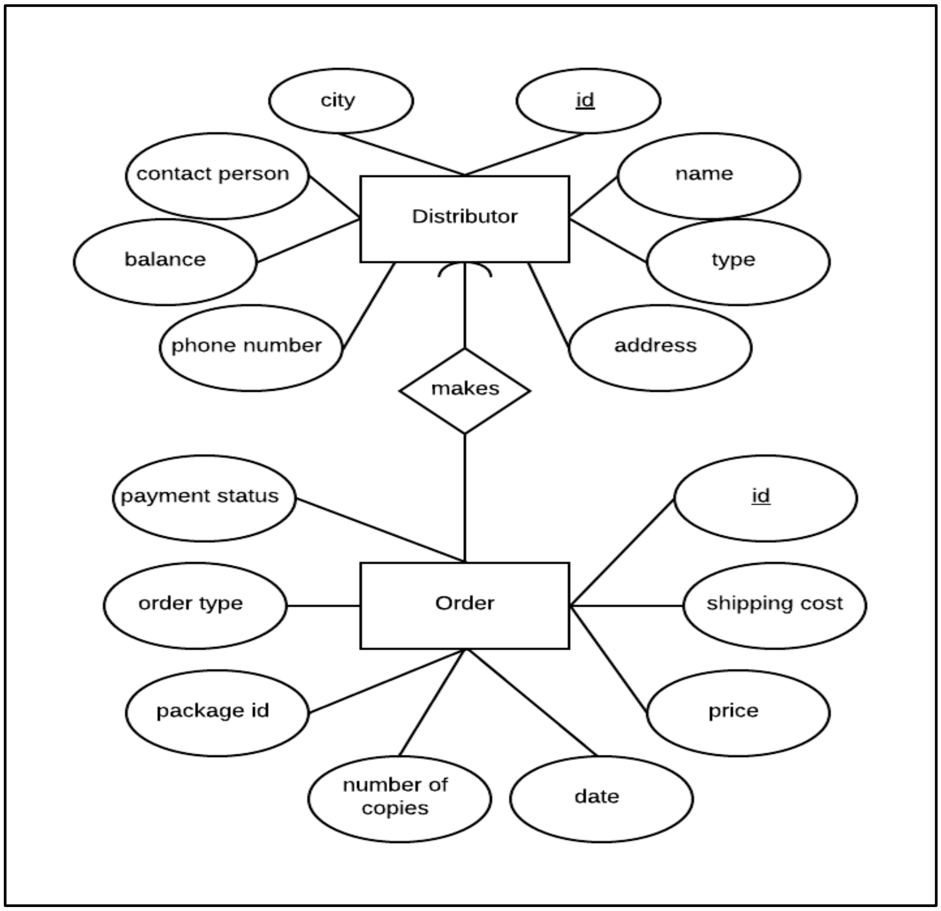
Issue's view

Editor’s view:



Editor’s view

Distributor's view:



Distributor's view

8. Description of Local E/R diagrams:

The proposed database system will be used by a publication house, distributors, administrator and editors for their day to day use.

- **Publication**:
    - The publication has a publication id, a primary key used to distinguish between two publications with the same name if there is any.
    - Along with the key, it will hold its type, periodicity, and topics it publishes.
    - Here publication will publish either books, articles or both so there is a relationship between publication, book and article entity.

- **Book**:
    - So, publication houses can publish books or articles in an issue so in order to distinguish there are divided by two different relationships.
    - ISBN will act as a primary key and each edition of a book has a unique id which is ISBN, so there is no need for a weak entity.

- **Issue**:
    - An article will be published in an issue that will be identified by an issue id.
    - Publication houses can publish many issues but an issue will be published by exactly one publication house so there is a many to one relationship.
- **Article**:
    - An article can have its own existence without any issue (see assumption #3)
    - An article will be part of many issues and vice versa. Let's take an example of a sportsman who can be part of an issue named sportsman and leadership and that's why there is many to many relationship.
- **Distributors**:
    - A distributor entity can have multiple orders, but an order can be associated with only one distributor.
    - The relationship named "makes" map a distributor with an order using distributor id and an order id.
- **Order**:
    - There can not be duplicate orders so each order will store order type i.e., book or article order, price, no. of copies.
- **Contributors**:
    - Publication house pays contributors which contains authors, editors with many to many relationships. This entity will hold attributes related to contributors such as their salary, payment information and its cycle like monthly, yearly.

9. Local Relational Schemas:

- **Admin view:**
  - Publication(id, title, typical_topics, type, periodicity)
  - Book(ISBN, author, date_of_creation, content, publication_date, edition_no, publication_id)
  - Issue(id, date_of_issue, publication_id)
  - Article(id, author, date_of_creation, content)
  - consistOf(issueID, ArticleID)
  - Contributor(id, name, designation, job_type, salary, payment_cycle_period)
  - WorksFor(contributorID, publicationID)
  - Pays(contributorID, publicationID, amount, payment_date)
  - Distributor(id, name, type, address, city, contact_person, balance, phone_number)
  - Order(id, shipping_cost, price, date, number_of_copies, packageID, order_type, payment_status, distributorID)

- **Author of book view:**
  - Publication(id, title, typical_topics, type, periodicity)
  - Book(ISBN, author, date_of_creation, content, publication_date, edition_no, publication_id)
  - Contributor(id, name, designation, job_type, salary, payment_cycle_period)
  - WorksFor(contributorID, publicationID)
  - Pays(contributorID, publicationID, amount, payment_date)

- **Author of Issue:**
  - Publication(id, title, typical_topics, type, periodicity)
  - Issue(id, date_of_issue, publication_id)
  - Article(id, author, date_of_creation, content)
  - consistOf(issueID, ArticleID)
  - Contributor(id, name, designation, job_type, salary, payment_cycle_period)
  - WorksFor(contributorID, publicationID)
  - Pays(contributorID, publicationID, amount, payment_date)

- **Editor of publication view:**
  - Publication(id, title, typical_topics, type, periodicity)
  - Book(ISBN, author, date_of_creation, content, publication_date, edition_no, publication_id)
  - Issue(id, date_of_issue, publication_id)

  - Article(id, author, date_of_creation, content)
  - consistOf(issueID, articleID)
  - Contributor(id, name, designation, job_type, salary, payment_cycle_period)
  - WorksFor(contributorID, publicationID)
  - Pays(contributorID, publicationID, amount, payment_date)

- **Distributers view:**
  - Distributor(id, name, type, address, city, contact_person, balance, phone_number)
  - Order(id, shipping_cost, price, date, number_of_copies, packageID, order_type, payment_status, distributorID)

10. Local Relational Documentation:

The **Admin view schema** consists of the information related to the administrator of the WolfCity publishing house. This schema consists of the details of the publication which contains columns like title, typical_topics, and periodicity. Then it contains information about the Book, Issues and Articles that are printed by the publication house. The Book entity has details related to identifying a particular book along with its edition number. Similarly issues and articles also have data that characterizes them like id, date of creation and content. We have added a consist of to link issues with their articles using their respective IDs. The pays relation connects the contributor and publication entities and has its own fields of amount and payment_date.

The **Author of book, Author of Issue** and the **Editor of publication view schemas,** belong to the Contributors view. For these users, we converted the entities from the E/R diagram into relation schema directly. This schema will be accessible only to the authors and the publishing house as the data will be pertaining to those two entities.

The **Distributors view schema** consists of the relations between the distributor and the orders (of editions of books and issues of articles) which are purchased by the distributor from the publishing house. This schema is accessible to the distributors and it consists of the details of the distributors referenced by their id along with their name, address, balance with the publishing house among other fields. It also consists of order history containing all current and past orders and their details and payments status.

Thus, the E/R diagram and the relational schemas accurately depict a complete view of the WolfPubDb.