WolfCity Publishing Management System

for WolfCity Publishing House

Project Report 1
CSC 540 Spring 2022

Andrew Abate, Chaitanya Patel, Morgan Chapman, Ilya Arakelyan
February 15, 2022

Assumptions

- Distributors are external to the database. They call the Distribution Team (sales) to order. A member of a Distribution team can place as many orders as they want. At the beginning of every month the Distribution Team generates an invoice for each distributor's outstanding balance.
- Following the narrative, the Distribution Team can place orders for book editions and publication issues (periodicals) based on a publication ID, but can search using other attributes.
- Following the narrative, the Distribution Team can change all Distributor's attributes, including its outstanding balance. Being a more frequent and explicitly requested task, the Distribution Team changing only the Distributor's balance is included as an additional (though redundant) operation.
- A publication must be a periodical issue or a book edition.
- The Production section of the API handles the creation, updating, and deletion of all publications. Overlapping publication operations were excluded from Editing/Publishing.
- Periodicity is specified by the Publication type attribute. Books are published by edition; all journals are published monthly; all magazines are published weekly.
- A publication has an overall topic of interest. Journals and magazines also include articles with their own topic, which can be the same as the publication topic or different. A book only has a single topic.
- Authors of books are associated with the book edition as a whole, not just the chapter(s) for which the author is responsible.
- Distributors are billed on a monthly schedule for their outstanding balance at the time of billing.
- There is no set limit to a distributor's outstanding balance or the number of orders a distributor can place.
- Prices are determined by publication type: all book editions are $10, all magazine and journal issues are $5.
- Shipping costs are calculated to be 5% of the order price.
- Staff editors and authors are expected to be paid on a biweekly schedule. Paychecks are submitted to invited editors and authors within 2 weeks of the publication or issue date.
- Article text is stored outside the database as a text file. The ArticleText attribute contains the filename for the text file.
- No chapters within a book edition may have the same title.
- No articles within a periodical issue may have the same title.
- Journalists are included in the authors table.
- Employees can't view the payment table. Their payment information will be on physical paychecks.
- Article creation dates are assumed to be the corresponding issue's release date.

1. Problem Description

We are creating a database system for a publishing company. The company needs to store and update information on their publication catalog, employees, distributors, and finances. Four primary user types will need to interact with the database: publishers (administrators), editors, distribution team (sales), and the financial team. They will perform a wide variety of operations: editing/publishing, production of new publications, handling of distributor orders and payments, and generation of financial reports.

Using a database for this problem would be the best way to keep track of all the information. The relations used in a database will help store and access data more quickly. The main advantages of this method include the ability to easily add, update, and delete data without any anomalies. A database also provides improved data integrity and better efficiency for users, resource management, costs, and growth. Additionally, users will be able to concurrently access all the data in the database.

2. Intended User Classes

Publishers: Manages the publishing house operations. Users in this class have access to all publication, distributor, and employee information in the database; can create new publications and modify/delete existing publications; manage staff and assign editors to work on publications; can add/modify distributors and send bills; and review monthly reports detailing the overall operation: total revenue (including detailed breakdown of publication orders) and total expenses (including payments to editing and writing staff).

Editors: Edit the publications before they can be published/distributed. They are responsible for looking at the database for publications assigned to them and editing them.

Distribution Team: Browse the catalog (publication DB). Place orders for publications. Can check order history. Can update the distributor's balance with the publishing house.

Financial Team: Can generate the monthly reports specified in the project narrative. Can bill the distributors based on their outstanding balance. Can send payments to employees. Can view all pending and completed invoices, orders, and payments.

3. Five Main "Things"

1. Publication Information: publication ID, publication title, publication type, publication topic, issue date, issue title, book edition number, book ISBN, book publication date, content for each publication
2. Distributor Information: distributor account number, name, type, address, city, phone number, contact person, outstanding balance
3. Order Information: order ID, number of copies, date to be produced, price, shipping costs
4. Financial Information (Payments and Invoices): payment check number, amount, submit date, date claimed by employee, invoice ID, amount, billing date, payment date
5. Employee Information: employee ID, name, role (editor or author), employment type (staff or invited), active status

4. Example Situations for Operations

Situation 1:

Description: A library (new distributor) becomes a customer and orders all children-themed books from the catalog.
Operations: a member of the Distribution Team *adds a new distributor* the library, *places an order* for all book items within the given topic

Situation 2:

Description: An editor is editing an upcoming issue of the weekly magazine *Wolf Weekly*. She has just received an article, which will become the cover story. She edits the issue by adding the article and changing the issue title.
Operations: Editor enters a new article and updates the issue

5. Application Program Interfaces

1. Editing and Publishing:

- assignEditorToPublication(employeeID, publicationID)
  return confirmation

- viewEditorResponsibilities(employeeID)
  returns list of publications for which editor is responsible

- addEditor(employeeID, name, type)
  returns confirmation

- updateEditor(employeeID, name, type, active)
  returns confirmation

- deleteEditor(employeeID)
  returns confirmation

- addAuthor(employeeID, name, type)
  returns confirmation

- updateAuthor(employeeID, name, type, active)
  returns confirmation

- deleteAuthor(employeeID)
  returns confirmation

2. Production:

- createBook(publicationID, title, type, topic, editionNumber, ISBN, publicationDate)
  return confirmation

- updateBook(publicationID, title, type, topic, editionNumber, ISBN, publicationDate)
  return confirmation

- deleteBook(publicationID)
  return confirmation

- createIssue(publicationID, title, type, topic, issueDate, issueTitle)
  return confirmation

- updateIssue(publicationID, title, type, topic, issueDate, issueTitle)
  return confirmation

- deleteIssue(publicationID)
  return confirmation

- addChapter(publicationID, chapterTitle)
  return confirmation

- updateChapter(publicationID, oldChapterTitle, newChapterTitle)
  return confirmation

- addArticle(publicationID, articleTitle, author, articleTopic)
  return confirmation

- updateArticle(publicationID, oldArticleTitle, articleTitle, author, articleTopic)
  return confirmation

- addArticleText(publicationID, articleTitle, articleText)
  returns confirmation

- updateArticleText(publicationTitle, articleTitle, articleText)
  returns confirmation

- getBook(authorName, publicationDate, topic)
  returns book information or NULL

- getBookByAuthor(authorName)
  returns book information or NULL

- getBookByDate(publicationDate)
  returns book information or NULL

- getBookByTopic(topic)
  returns book information or NULL

- getArticle(authorName, creationDate, topic)
  returns article information or NULL

- getArticleByAuthor(authorName)
  returns article information or NULL

- getArticleByDate(creationDate)
  returns article information or NULL

- getArticleByTopic(topic)
  returns article information or NULL

- payEmployee(employeeID, checkNumber, amount, submitDate)
  returns confirmation

- acceptPayment(employeeID, checkNumber, claimDate)
  returns confirmation

- trackPayment(employeeID, checkNumber)
  returns claimDate of employee or NULL

3. Distribution:

- addDistributor(DistAccountNum, Name, Type, Address, City, PhoneNumber, Contact, Balance)
  return confirmation

- deleteDistributor(DistAccountNum)
  return confirmation

- updateDistributor(DistAccountNum, Name, Type, Address, City, PhoneNumber, Contact, Balance)
  return confirmation

- placePublicationOrder(PublicationID, NumCopies, ProduceByDate, DistAccountNum)
  return OrderID

- billDistributor(InvoiceID, DistAccountNum)
  return confirmation

- changeDistributorBalance(DistAccountNum, Amount)
  return NewBalance

4. Reports:

- generateReport(month)
  return report for given month

- totalMonthlyPubSold(publicationTitle, distributor, ProduceByDate)
  return total price and number of copies of the publication sold per distributor

- totalMonthlyPubRev(ProduceByDate)
  return total monthly revenue of the publishing house

- totalMonthlyExpense(expenseType, month)
  return total expenses of the publishing house depending on
  the type of expense (shipping cost, salaries, Total)

- totalDistributors()
  return calculated total current number of distributors

- totalCityRev(city)
  return calculated total revenue(since inception) per city

- totalDistRev(DistAccountNum)
  return calculated total revenue(since inception) per distributor

- totalLocRev(address, city)
  return calculated total revenue(since inception) per location

- totalPaymentsTime(employeeType, timePeriod)
  return total payments to the editors or authors, per time period(number of Months).

- totalPaymentsType(employeeType, workType)
  return total payments to the editors or authors, per work type (book authorship, article authorship, or editorial work).

6. Description of Class Views

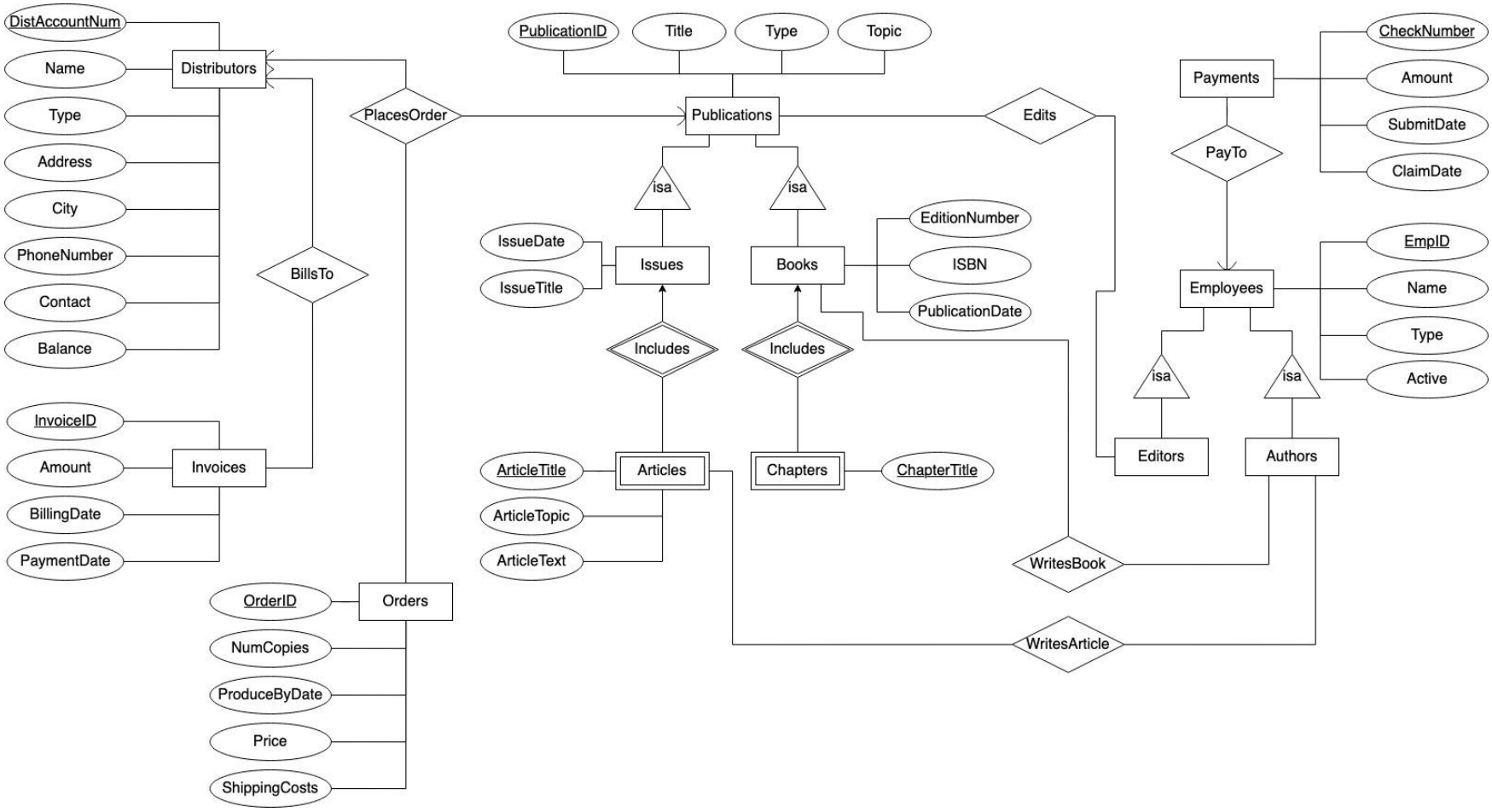
Publisher class view: Users can access all information in the database. Publishers can view all publication information, distributor accounts, order information, and all employee information including payments. Users can access the data individually, or obtain a summary view of the publishing house operations as a monthly report.

Editor class view: Users can view the information on the publications they have been assigned to including content (articles/chapters) and author information.

Distribution Team class view: Distribution Team can access the publication catalog (attributes relevant to a publication), distributor attributes, including the outstanding balance, and order history.

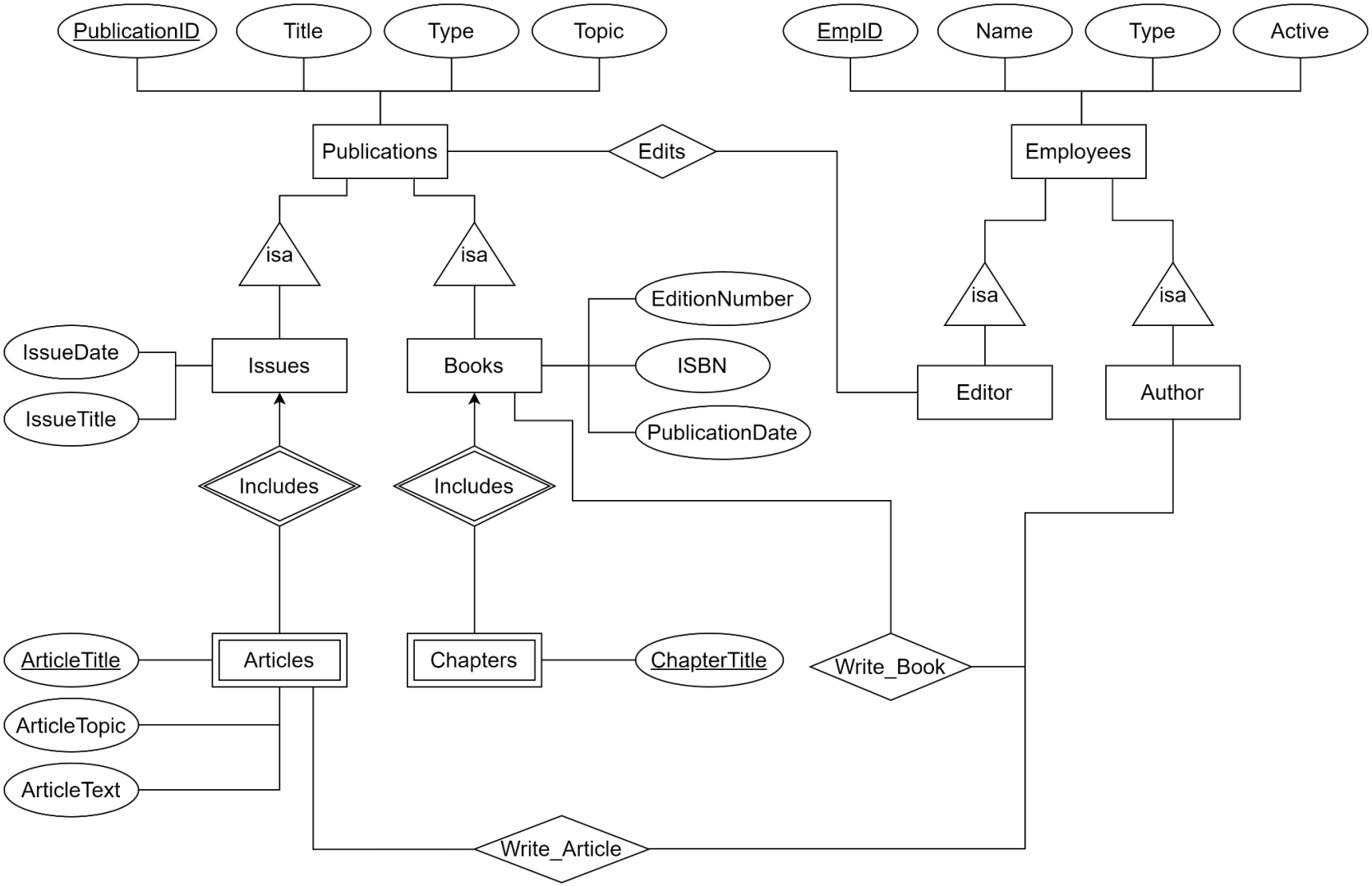
Financial Team: Users can access everything needed to generate the monthly reports and pay employees. This includes all invoices, orders, distributors, payments, and employees.

7. Local E/R Diagram[1]: Publishers

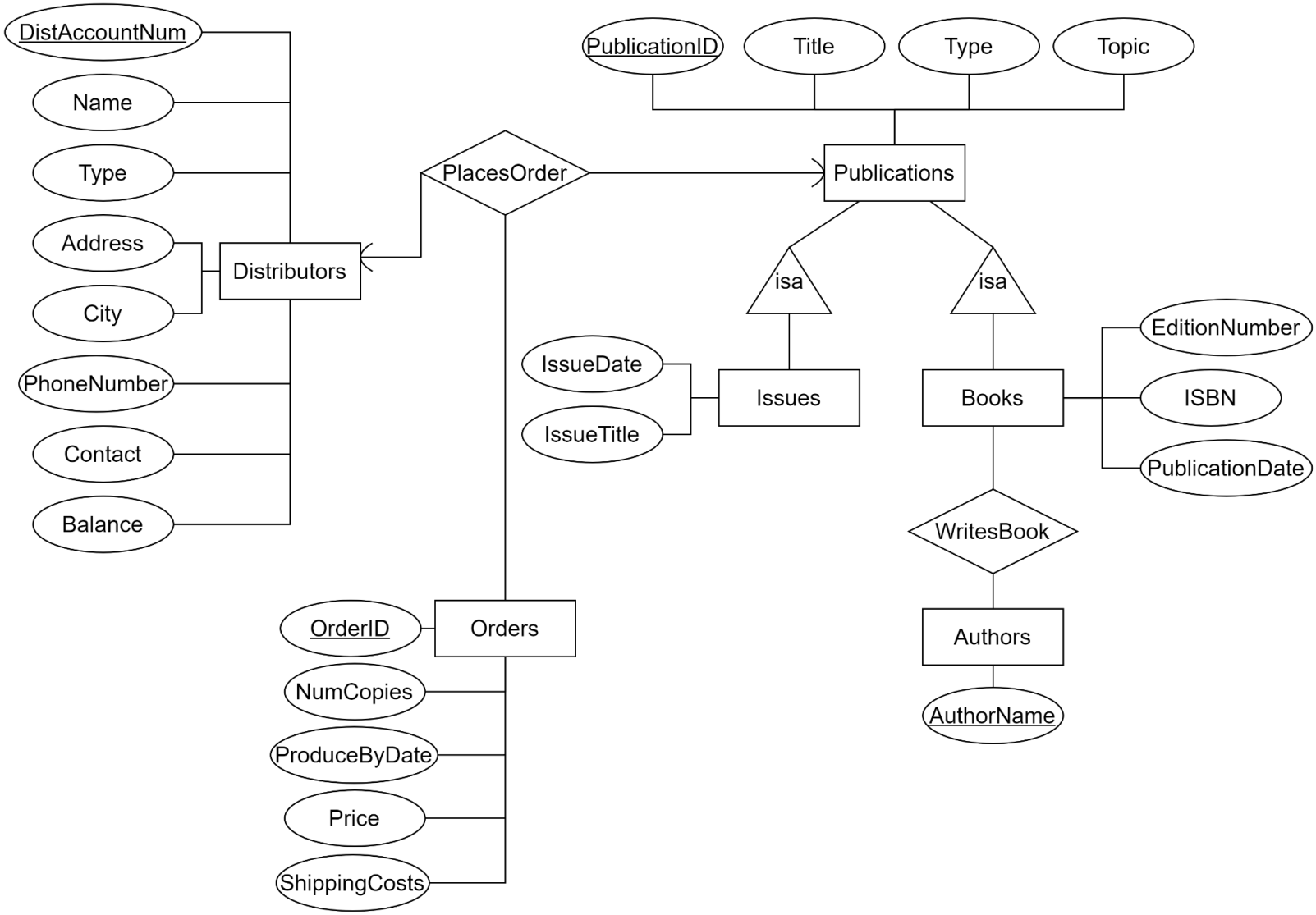


[1] We provide data format in section 9, Local Relational Schema

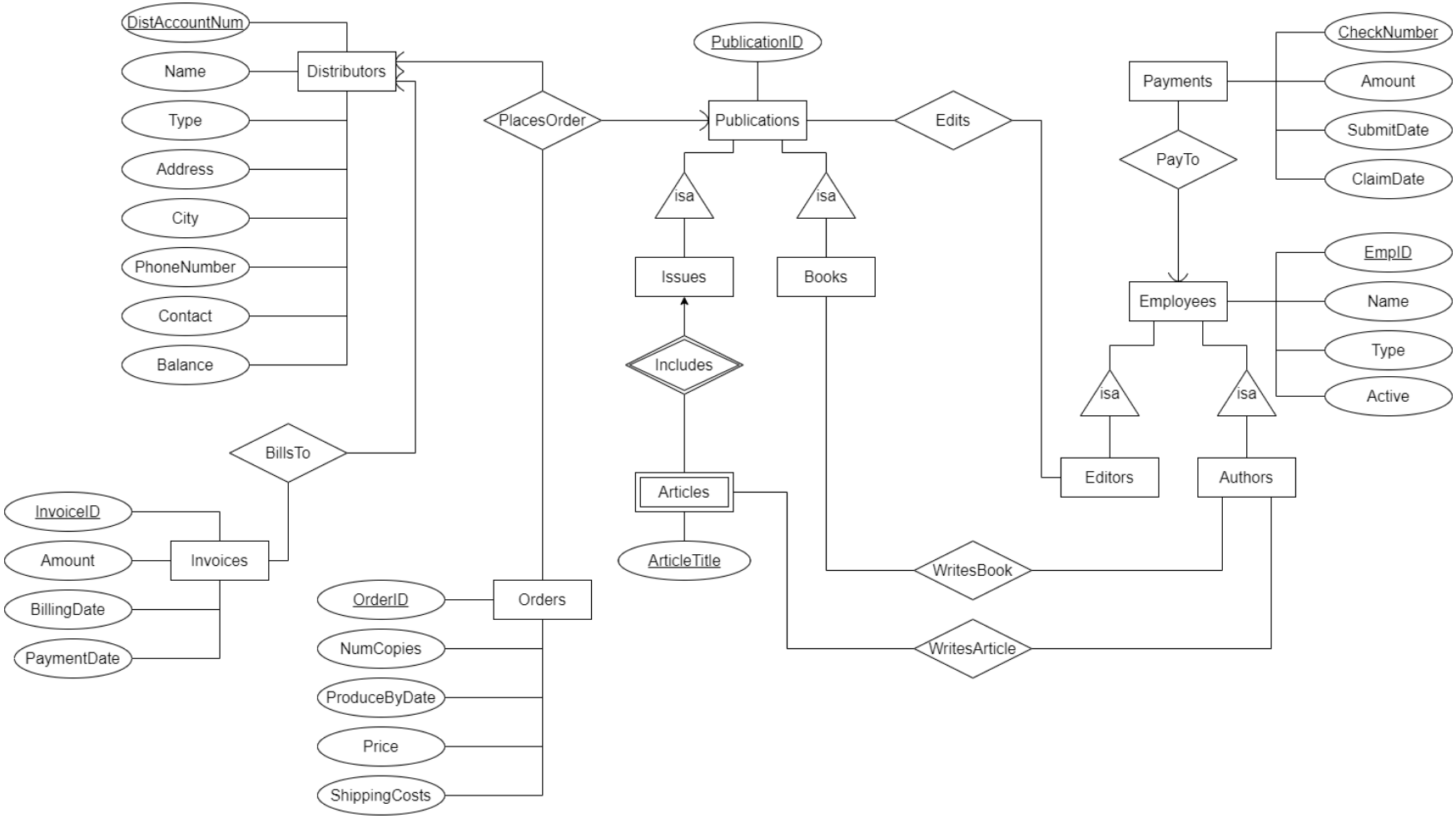
Local E/R Diagram: Editors



Local E/R Diagram: Distribution Team



Local E/R Diagram: Financial Team



8. Description of Local E/R Diagrams

- **Publications** is the root of the publication information hierarchy and contains attributes for publication ID, publication title, type of publication (book, journal, or magazine) and a single topic. Publications are uniquely identified by their publication ID. A publication type indicates its periodicity, as suggested in the narrative and assumptions.
- Publications are divided into subclasses of **Issues** (for journal and magazine periodicals) and **Books** (for book editions) via is-a relationships. This is a disjoint classification: a publication can be either an Issue or a Book, not both.
- **Issues** contain attributes for issue date and issue title. The publication ID, publication title, type, and topic attributes are inherited from Publications. Issues are uniquely identified by the inherited publication ID.
- **Books** contain attributes for edition number, ISBN, and publication date. The publication ID, publication title, type, and topic attributes are inherited from Publications. Books are uniquely identified by the inherited publication ID.
- Issues **include** articles. **Articles** contain attributes for article title, article topic, and article text. An article may only be included in a single issue (many-to-one relationship to Issue). As described in the assumptions, an article's creation date is considered to be the same as the issue date of the issue to which it belongs - a separate article date attribute is not needed. Articles are a weak entity set because they are uniquely identified by publication ID from Issues and article title. **Includes** represents this supporting relationship. This design allows for the reasonable situation of different articles with the same title appearing in different Issues.
- Books **include** chapters. **Chapters** contain a chapter title attribute. Chapters are weak entity sets because they are uniquely identified by publication ID from Books and chapter title. **Includes** represents this supporting relationship. This design allows the reasonable situation of different books containing chapters with the same title. If no specific chapter title exists, a chapter number can be included as the title.
- **Employees** contain information on the editors and authors working for the publishing house, including a unique employee ID number, name, employment type (staff or invited contributor), and active status indicator. The employment type and active/not-active status attributes are used for determining payments.
- Employees are subdivided into **Editors** and **Authors**, each identified by their employee ID attribute inherited from Employees.
- Editors **edit** one or more publications. A publication may have more than one editor. The **Edits** relationship is assigned at the publication level as editors can be assigned to edit books and periodical publications.
- Authors can **write** book editions and/or articles for issues. An article or book edition may have more than one author. Multiple Writes (**WritesArticle** and **WritesBook**) relationships represent the relationships from Authors to Articles and Books, respectively. Note: for book editions authorship is attributed at the Books level instead of Chapters.

- **Payments** are uniquely identified by a check number and contain information about the amount, date of submission to the employee, and date of claim by the recipient. A Payment is made to a single employee through the **PayTo** relationship and is associated with their employee ID number representing a referential integrity constraint. A payment's claim date attribute should be updated when claimed by the recipient.
- **Distributors** contain all distributor information including a unique distributor account number, name, type (wholesale distributor, bookstore, or library), address, city, phone number, contact name, and outstanding balance.
- Distribution team can **place orders** for issues or book editions. Orders are placed for a single publication at a time. However, the distribution team may place multiple orders for the same target delivery date or multiple (separate) orders for the same publication. Orders are made for a specific publication ID, which identifies a unique issue or book edition. Therefore, **PlacesOrder** is represented as a three-way many-to-one relationship between each order, a corresponding unique distributor, and a publication.
- **Orders** for publications contain a unique order ID number, number of publication copies, date by which an order should be filled, price, and shipping costs. As described in the Assumptions, order prices are determined by the type of publication and shipping costs are charged as a percentage of the order price.
- **Invoices** contain information on bills sent to distributors. Invoices are uniquely identified by their invoice ID number and contain attributes for amount, billing date, and payment date.
- Invoices are **billed to** distributor accounts with an invoice amount determined by the distributor's outstanding balance on the given billing date. As invoices are not determined by particular **Orders**, no connection between **PlaceOrder** and **BillTo** is required. The **BillsTo** relationship is represented as a many-to-one relationship with referential integrity, as each invoice is associated with exactly one distributor account.
- The local E/R diagram for the **Publisher** class includes all publication, distributor, order invoice, employee, and payment information and connecting relationships. These are required to provide monthly summary reports reviewed by the Publisher.
- The local E/R diagram for the **Distribution Team** class includes all distributor and order information in addition to the key publication information required to handle orders.
- The local E/R diagram for the **Editor** class includes all publication and employee information and relationships.
- The local E/R diagram for the **Financial Team** class includes all distributor, order, and invoice information and relationships. All employee and payment information is also accessible. Only the key publication and article information is needed for billing and payment operations.

9. Local Relational Schemas

Publisher View:

Publication:

- Publications(PublicationID:integer, Title:string, Type:string, Topic:string)
- Issues(PublicationID:integer, IssueDate:date, IssueTitle:string)
- Articles(PublicationID:integer, ArticleTitle:string, ArticleTopic:string, ArticleText:string)
- Books(PublicationID:integer, EditionNumber:integer, ISBN:integer, PublicationDate:date)
- Chapters(PublicationID:integer, ChapterTitle:string)

Staff:

- Employees(EmpID:integer, Name:string, Type:string, Active:boolean)
- Editors(EmpID:integer)
- Authors(EmpID:integer)
- Edits(PublicationID:integer, EmpID:integer)
- WritesBook(PublicationID:integer, EmpID:integer)
- WritesArticle(PublicationID:integer, ArticleTitle:string, EmpID:integer)
- Payments(EmpID:integer, CheckNumber:integer, Amount:float, SubmitDate:date, ClaimDate:date)

Distribution:

- Distributors(DistAccountNum:integer, Name:string, Type:string, Address:string, City:string, PhoneNumber:string, Contact:string, Balance:float)
- Orders(DistAccountNum:integer, PublicationID:integer, OrderID:integer, NumCopies:integer, ProduceByDate:date, Price:float, ShippingCosts:float)
- Invoices(DistAccountNum:integer, InvoiceID:integer, Amount:float, BillingDate:date, PaymentDate:date)

Editor View:

Publication:

- Publications(PublicationID:integer, Title:string, Type:string, Topic:string)
- Issues(PublicationID:integer, IssueDate:date, IssueTitle:string)
- Articles(PublicationID:integer, ArticleTitle:string, ArticleText:string, Topic:string)
- Books(PublicationID:integer, EditionNumber:integer, ISBN:integer, PublicationDate:date)
- Chapters(PublicationID:integer, Title:string)

Staff:

- Employees(EmpID:integer, Name:string, Type:string, Active:boolean)
- Editors(EmpID:integer)
- Authors(EmpID:integer)
- Edits(PublicationID:integer, EmpID:integer)
- WritesBook(EmpID:integer, PublicationID:integer)
- WritesArticle(EmpID:integer, PublicationID:integer, ArticleTitle:string)

Distribution Team View:

Publication:

- Publications(PublicationID:integer, Title:string, Type:string, Topic:string)
- Issues(PublicationID:integer, IssueDate:date, IssueTitle:string)
- Books(PublicationID:integer, EditionNumber:integer, ISBN:integer, PublicationDate:date)
- Authors(AuthorName:string)
- WritesBook(PublicationID:integer, AuthorName:string)

Distribution:

- Distributors(DistAccountNum:integer, Name:string, Type:string, Address:string, City:string, PhoneNumber:string, Contact:string, Balance:float)
- Orders(DistAccountNum:integer, PublicationID:integer, OrderID:integer, NumCopies:integer, ProduceByDate:date, Price:float, ShippingCosts:float)

Financial Team View

Publication:

- Publications(PublicationID:integer)
- Books(PublicationID:integer)
- Issues(PublicationID:integer)
- Articles(PublicationID:integer, ArticleTitle:string)

Staff:

- Employees(EmpID, Name, Type, Active)
- Editors(EmpID:integer)
- Authors(EmpID:integer)
- Edits(PublicationID:integer, EmpID:integer)
- WritesBook(EmpID:integer, PublicationID:integer)
- WritesArticle(EmpID:integer, PublicationID:integer, ArticleTitle:string)
- Payments(EmpID:integer, CheckNumber:integer, Amount:float, SubmitDate:date, ClaimDate:date)

Distribution:

- Distributors(DistAccountNum:integer, Name:string, Type:string, Address:string, City:string, PhoneNumber:string, Contact:string, Balance:float)
- Orders(DistAccountNum:integer, PublicationID:integer, OrderID:integer, NumCopies:integer, ProduceByDate:date, Price:float, ShippingCosts:float)
- Invoices(DistAccountNum:integer, InvoiceID:integer, Amount:float, BillingDate:date, PaymentDate:date)

10. Description of Local Relational Schemas

Entity Sets with Hierarchy Relations:

- Books and Issues(periodicals) are in a is-a relationship with publication in order for them to share the same attributes and PublicationID is used as the key. Editor and Author also are in a is-a relationship with Employee and have EmpID as the key. They were made into hierarchy relations to avoid redundancy and save table space from an E/R viewpoint.

Entity Set Translation:

- All entity sets were converted into schemas by including all attributes and indicating the identifying attributes as primary keys.
    - Publications, Distributors, Orders, Invoices, Employees, Payments
- Entity sets representing subclasses through is-a relationships were converted into schemas by including the inherited primary key and any subclass-specific attributes.
    - Issues, Books, Editors, Authors

Weak Entity Set Translation:

- Weak entity sets were converted into schemas by including all attributes in addition to the key attributes from the supporting entity set. Supporting relationships from weak entities were not converted into schemas and there were no additional relationship attributes.
    - Articles, Chapters

Many-to-One Relationship Translation:

- For entity sets involved in a many-to-one relationship, the entity set representing the "many" component must contain exactly one entity from the connected set(s). In these situations, the key for the "one" side of the relationship was included in the schema for the "many" side entity set. The connected relationships were not converted to schemas to avoid redundancy and there were no additional relationship attributes to include.
    - Orders, Invoices, Payments
    - Note: The Orders entity is involved in the 3-way many-to-one relationship PlacesOrder. Thus, it includes keys from two related entity sets: Distributors and Publications.

Many-to-Many Relationship Translation:

- Relationships that represent many-to-many connections of entity sets were converted into schemas by including keys from each connected entity set. Any attributes specific to the relationship should also be included, but no such situation was represented in our diagram.
    - Edits, WritesArticle, WritesBook