CSC 540 DATABASE MANAGEMENT CONCEPTS AND SYSTEMS

WOLFCITY PUBLICATION HOUSE

Project Report 1

Team Members (Q):
Alisha Shahane (asshahan)
Shruti Kangle (sskangle)
Poorva Kulkarni (pnkulkar)

Assumptions

a. Employees:
    i. The employees of the publication house are divided into three types based on the roles they play. They are:
        1. Writers
        2. Accountants
        3. Administrators
    ii. Each writer is either a staff member who is paid on a particular date every month (periodically) or an invited writer who is paid only once.
    iii. A writer can either be an author, editor or journalist and he/she has access to only the edition or issue he/she has been assigned.
b. Publications:
    i. The books, magazines or journals are identified by their ISBN.
    ii. All editions or issues of a book or magazine have the same ISBN.
    iii. The inventory of the publication house keeps records of all the editions of a given publication.
c. Order:
    i. We maintain the status of every order, and it can be
        1. Accepted
        2. Processing
        3. Completed
        4. Discarded
    ii. A single order can contain only one issue/edition of a book/magazine. An order is completed only when all its requirements are met before the expected delivery date. After the expected delivery date is passed, the order can no longer be completed, and it is discarded.
    iii. Shipping cost is equal to a fraction of (i.e. 10%) of the number of copies in any order.
d. Miscellaneous:
    i. The system's scope is limited to a single publication house.
    ii. Any add/update/delete operation will take emp_id as one of the parameters and will be performed only if that is a permitted operation for that employee type.
    iii. No transaction can be deleted or updated. A transaction can only be inserted.
    iv. After a given time interval, all the requests in the 'Requests' table are fulfilled, and the corresponding copies are updated in the inventory. The entry from the 'Requests' table is removed after that.

1.1 Problem Statement

The WolfPubDB is designed for the general operations of WolfCity publishing house. The database is designed from the viewpoint of the employees of the publishing house. It is capable of handling all the day-to-day operations that concerns a publishing house. All the functionality of the database revolves around publishing new content, managing orders and maintaining books for the publishing house.

A writer (a type of employee) has the job of writing and/or editing new content. Once any new literature is written, editors are assigned to it. Only after the content is edited, it is added to publishing house inventory. Writers have the ability to update selective content over the period of time.

An accountant is responsible for managing financial matters for the publishing house. Tracking payments from distributors and handling salaries for all employees is an accountant's main task. The transaction database, which contains the most sensitive information and is crucial for the generating reports, is accessible only to the accountants.

Admin handles the daily activities like updating inventory and managing orders. Inventory updating is done once the admin is notified by the writers. Order management includes activities like accepting orders, requesting inventory, maintaining status etc.

Why are databases better than files?

a. Standardizing Data: Applying the same format across all applications is easier in database systems, whereas file systems have scattered isolated data.
b. Concurrent Data Access: In the scenario where there are many roles (authors, distributors, administrators) simultaneously trying to access and modify the data, files would be inefficient. File copies need to be maintained at every individual level and consolidating them into a consistent copy would be a difficult task. Also, editing a particular record in multiple files would cause a latency. Hence, databases are well-suited for concurrent data access.
c. Security: Databases manage the security of an application by not allowing restricted users to modify, add or delete any information they are not authorized to.
d. Ease of Data Access: It becomes easy to access any random attribute of any random tuple in a database. This is usually a difficult task to retrieve a specific data in the traditional file system.
e. Data Redundancy: File systems may have redundant data causing duplication. These multiple copies of the same data are difficult to update and may cause inconsistency. Databases ensure that the stored data is unique and further no delete/update anomalies occur.
f. Resilience: Since database systems keep the backup of data, it is easier to do a full recovery of data in case of a failure.

1.2 Intended Classes of Users

a. Writer: Writer creates literature and receives salary from publishing house.
b. Accountant: Is responsible for paying salaries of all employees of WolfCity and collecting the payments from distributors. Maintains monetary transactions of employees and distributors.
c. Admin: Receives the order placed by a distributor. Manages the order fulfilment through the inventory.
d. Distributor: Places order for a certain number of copies of a publication and makes the corresponding payment to the publishing house.

1.3 Five main things

a. Employees (writers, accountants, admins):

    Employee ID, employee name, email, salary, employee type

b. Inventory:

    ISBN, edition, price, available copies, booked copies, issue date

c. Literature Information:

    ISBN, edition, literature text, genre, title, literature type, periodicity

d. Transactions:

    Transaction ID, Client ID, amount, date

e. Orders:

    Order ID, edition, ISBN, distributor ID, copies, delivery data, amount, status

1.4 Realistic situations using tasks and operations in narrative

Situation a:

In order to publish the next edition of a book written by an invited author, the “Editing and Publishing” task adds this book in the system, assigns the editor to work on it and after editing, the editor notifies the admin to send it for publishing and update the inventory.

Situation b:

A distributor wants to place an order with WolfCity publishing house. The “Distribution” task adds the distributor details to the database. The admin discusses the requirements with the distributor and places an order of 1000 copies of an edition of a book to be delivered in 3 weeks.

1.5 Application Program Interfaces:

- Editing and Publishing:

Operation	Input	Output
addLiterature()	ISBN, edition, emp_id, title, lit_type, periodicity, text	Confirmation
updateLiterature()	ISBN, edition, emp_id, title, lit_type, periodicity, text	Confirmation
deleteLiterature()	emp_id, ISBN, edition	Confirmation
assignWriterToLiterature()	ISBN, edition, emp_id	Confirmation
unassignWriterFromLiterature()	ISBN, edition, emp_id	Confirmation
viewAssignedLiterature()	emp_id	ISBN, edition, title, genre, lit_type, periodicity, text of all literatures assigned to editor

- Production of a book edition or of an issue of a publication:

Operation	Input	Output
findBooks()	search_key(genre/ issue_date /emp_name) search value	ISBN, Title, Edition of all literatures that match the search criteria
addToInventory()	emp_id, ISBN, edition, issue_date, available_copies, booked_copies, price	Confirmation
updateInventory()	emp_id, ISBN, edition, issue_date, available_copies, booked_copies, price	Confirmation
deleteFromInventory()	emp_id, ISBN, edition	Confirmation
requestCopies()	emp_id, ISBN, edition, requested_copies	request_id or NULL if error
deleteRequest()	request_id	Confirmation

- Distribution:

Operation	Input	Output
addDistributor()	name, dist_type, city, location, street_address, contact, person_of_contact, balance	dist_id or NULL if error
updateDistributor()	dist_id, name, dist_type, city, location, street_address,, contact, person_of_contact, balance	Confirmation
deleteDistributor()	dist_id	Confirmation
createOrder()	dist_id, ISBN, edition, amount, exp_del_date, status, num_copies	order_id or NULL if error
updateOrder()	order_id, dist_id, ISBN, edition, amount, exp_del_date, status, num_copies	Confirmation
deleteOrder()	order_id	Confirmation

- Reports:

Operation	Input	Output
getDistributorData()	month	number and total price of copies of each publication bought per distributor for that month
getTotalRevenue()	-	total revenue of the publishing house
getTotalExpenses()	-	total expenses of the publishing house
getCurrentDistributors()	-	total number of distributors whose order status is “Processing”
calcTotalRevenue()	-	Calculates total revenue (since inception) per city, per distributor, and per location.
calcTotalPayments()	-	Calculate total payments to the editors and authors, per time period and per work type

- Manage Employees and Transactions:

Operation	Input	Output
addWriter()	name, email, salary, emp_type, writer_type, writer_status, pay_cycle, day_of_pay	emp_id or NULL if error
addOtherStaff()	name, email, salary, emp_type	emp_id or NULL if error
updateWriter()	emp_id, name, email, salary, emp_type, writer_type, writer_status, pay_cycle, day_of_pay	Confirmation
updateOtherStaff()	emp_id, name, email, salary, emp_type	Confirmation
deleteEmployee()	emp_id	Confirmation
addTransaction()	date, client_id, amount	trans_id or NULL if error

1.6 Views of User Interfaces

a. Writer:
   - Writers create literature and have writer types as author, editor or journalist.
   - Writers are subdivided into staff writers and invited (guest) writers which are denoted by 'writer_status' attribute.
   - The staff writers are paid a salary every month, while guest writers are given a payment only once.
   - Writers can view and modify only the literature they contributed to. They can update information such as title, literature type, genre, periodicity, text of their own compositions.

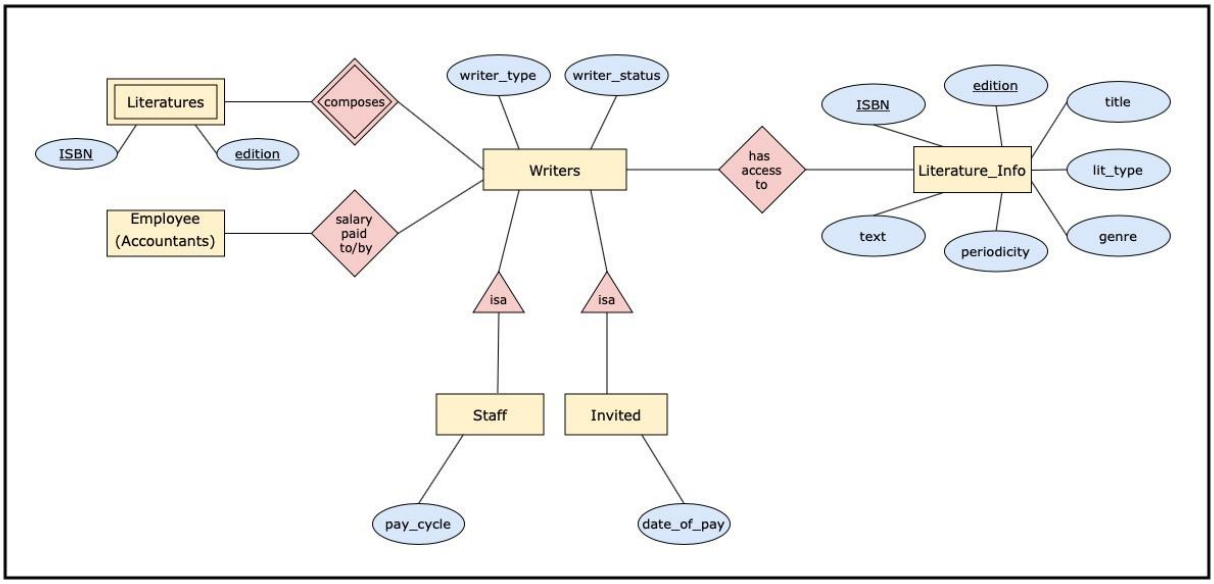
b. Accountant:
   - Accountant maintains the monetary transactions occurring in the system.
   - He is responsible for keeping track of payments made, the date of payment, balance and client ID which denotes with which party is the transaction being made with.
   - An accountant pays salaries to all the employees of WolfCity publication house.
   - He also is concerned with receiving payments from distributors corresponding to each order.

c. Admin:
   - The administrator has access to the orders table and the inventory of the publishing house.
   - He can add or remove the number of copies of each literary work in the inventory.
   - The admin interacts with writers to check if they are done composing a literary work and adds copies of the new works to the inventory.
   - He is also responsible for managing the orders by checking the amount of copies of a given literary work available and fulfilling the order.
   - If the number of copies are found to be inadequate, he requests more copies to be printed and adds them to the inventory.

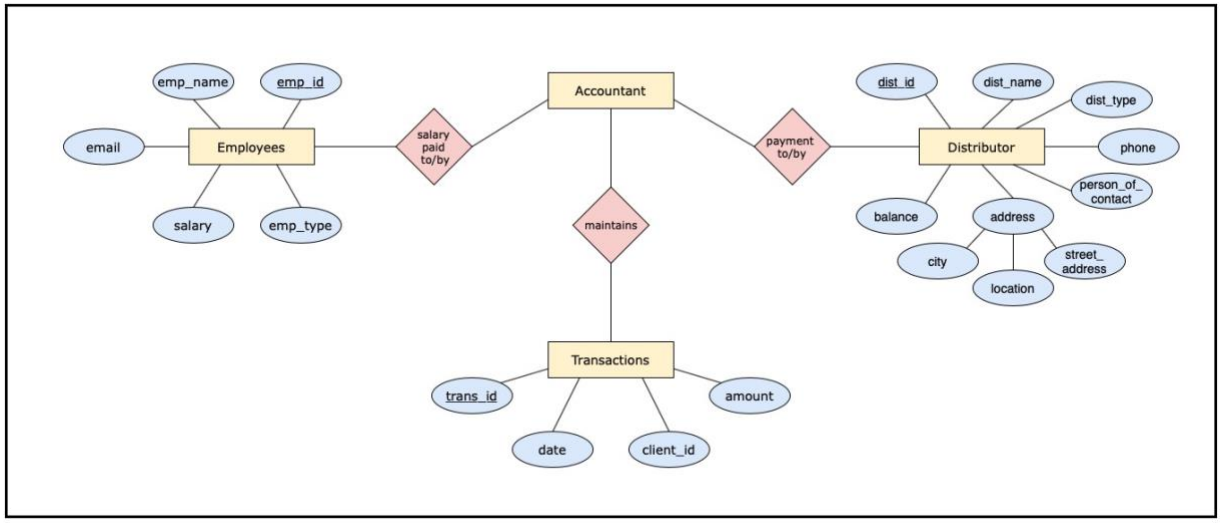
d. Distributor:
   - Distributors are identified by their unique distributor ID and place an order of books to the publishing house.
   - Distributor can view and modify the orders table.
   - He can manipulate the number of copies of each book, ISBN and edition which he wishes to order.
   - Distributor makes a payment to the accountant of the publishing house.

1.7 Local ER diagrams

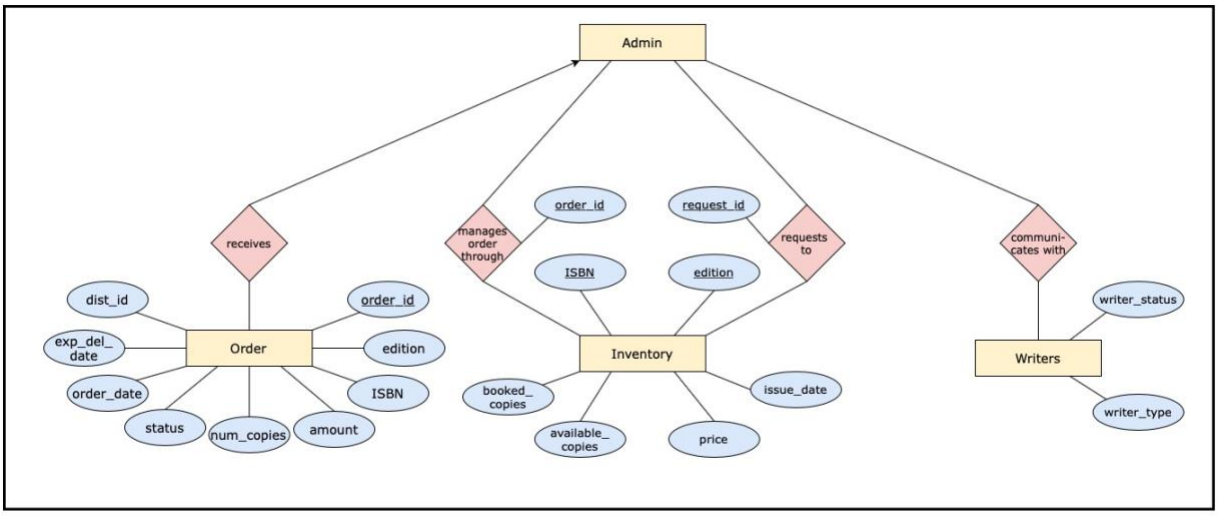
a. Writer



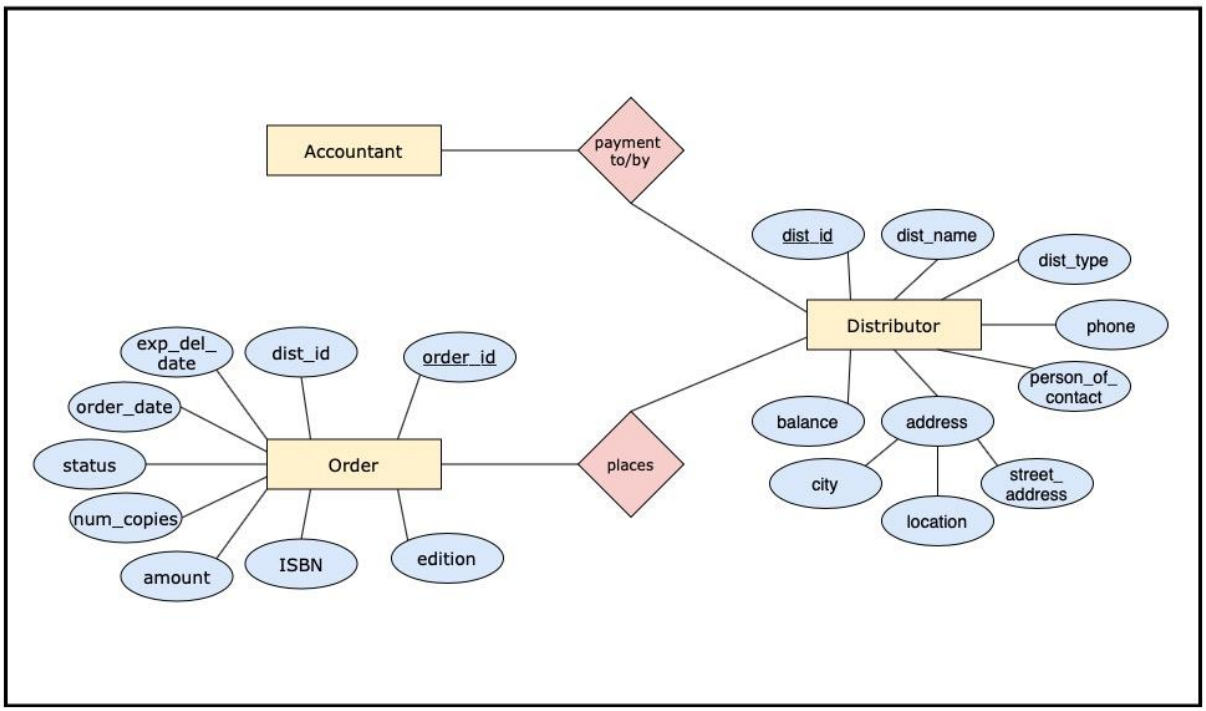
b. Accountant



c. Admin



d. Distributor



1.8 Documentation of ER Diagrams

Our database design focuses on operations of writers, accountants, admins and distributors.

a. Writer: A writer is a staff or an invited member.
    i. composes: A literature is composed by an author. Literature is a weak entity as it needs to be mapped to a writer.
    ii. has access to: A writer can access all the literature he has contributed to.
    iii. salary paid by: Be it staff or invited, the salaries are handled by the accountant.
b. Accountant:
    i. salary paid to: All the employees of the publishing house are handed salary by the accountant.
    ii. maintains: Only the accountant manages the transactions database.
    iii. payment by: Accountant handles payments done by the distributor.
c. Admin:
    i. receives: Admin receives every order and maintains its status.
    ii. communicates with: A writer notifies an admin once he/she has edited a publication.
    iii. manages order through: Admin has access to the inventory database, and he performs activities like keeping track of available and booked copies to manage orders.
d. Distributor:
    i. places: The distributor places an order with the publishing house which contains their requirements of any literary work.
    ii. payment to: The distributor settles the balance for the placed orders with the accountant of the publishing house.

1.9 Local Relational Schemas

Writer:

- Writer(emp_id, writer_type, writer_status, pay_cycle, date_of_pay, name, email, salary, emp_type)
- hasAccessTo(emp_id, ISBN, edition)

Admin:

- Admin(emp_id, name, email, salary, emp_type)
- Inventory(ISBN, edition, issue_date, price, available_copies, booked_copies)
- managesOrderThrough(emp_id, ISBN, edition, order_id)
- communicatesWith(emp_id)
- receivesOrder(order_id, emp_id)
- requestsTo(emp_id, request_id)

Accountant:

- Accountant(emp_id, name, email, salary, emp_type)
- Employees(emp_id, emp_name,email,salary,emp_type)
- Transactions(trans_id, date, client_id, amount)
- maintains(emp_id, trans_id)
- salaryPaidTo(emp_id, dist_id)
- paymentTo(dist_id, emp_id)

Distributor:

- Distributor(dist_id, dist_name, dist_type, phone, city, location, street_address, person_of_contact, balance)
- Order(order_id, dist_id, edition, ISBN, exp_del_date, order_date, status, num_copies, amount)
- places(order_id, dist_id)

1.10 Local Schema Documentation

- Each entity set was made into a relation with the same set of attributes.
- Relationships were replaced by a relation whose attributes are the keys for the connected entity sets.
- The E/R viewpoint was used to convert entity sets to relational schemas.
- Each type of employee inherits common attributes from the Employee table and also some attributes and relations specific to a given type, therefore they are represented by “is a” relation with the Employee table.
- The Literature entity is a weak entity since each entry in the table can be uniquely identified by its primary keys (ISBN, edition) and also the primary key of the Employee table(emp_id).
- In order for the relation “manage orders through” to hold between Admin and Inventory entities, we require the attribute “order_id” on the relation to uniquely identify the order specific information in the inventory.
- The admin needs to request the inventory for more copies of a given literary work and hence we require an attribute called “request_id” on the relation “requestsTo”.
- The address of the distributor is a composite attribute since it contains the attributes: city, location, street_address.
- “Receives” is a one-many relation since one Admin can receive many Orders.
- “Places” is a one-many relation since one Distributor can place many Orders and any order can only be placed by a single distributor.