Mini Project : Steps (2)-(4)

Spring 2025 - CMPT 354

Paul Huang (301622331) & Misha Zoubarev (301402619)

(2) Project Projections

LibraryItem(ItemID, Title, Author, Type, Status, Audience, DateAdded)
- **PK**: ItemID

User(UserID, Name, Phone, Address, Membership, AmountOwed)
- **PK**: UserID

Borrows(TransactionID, BorrowDate, DueDate, ReturnDate, UserID, ItemID)
- **PK**: TransactionID
- **FK**: UserID -> User(UserID) ***and*** ItemID -> LibraryItem(ItemID)

Fine(FineID, Amount, Status, UserID)
- **PK**: FineID
- **FK**: UserID -> User(UserID)

Event(EventID, Name, Description, RecommendedAudience, Location, DateTime)
- **PK**: EventID

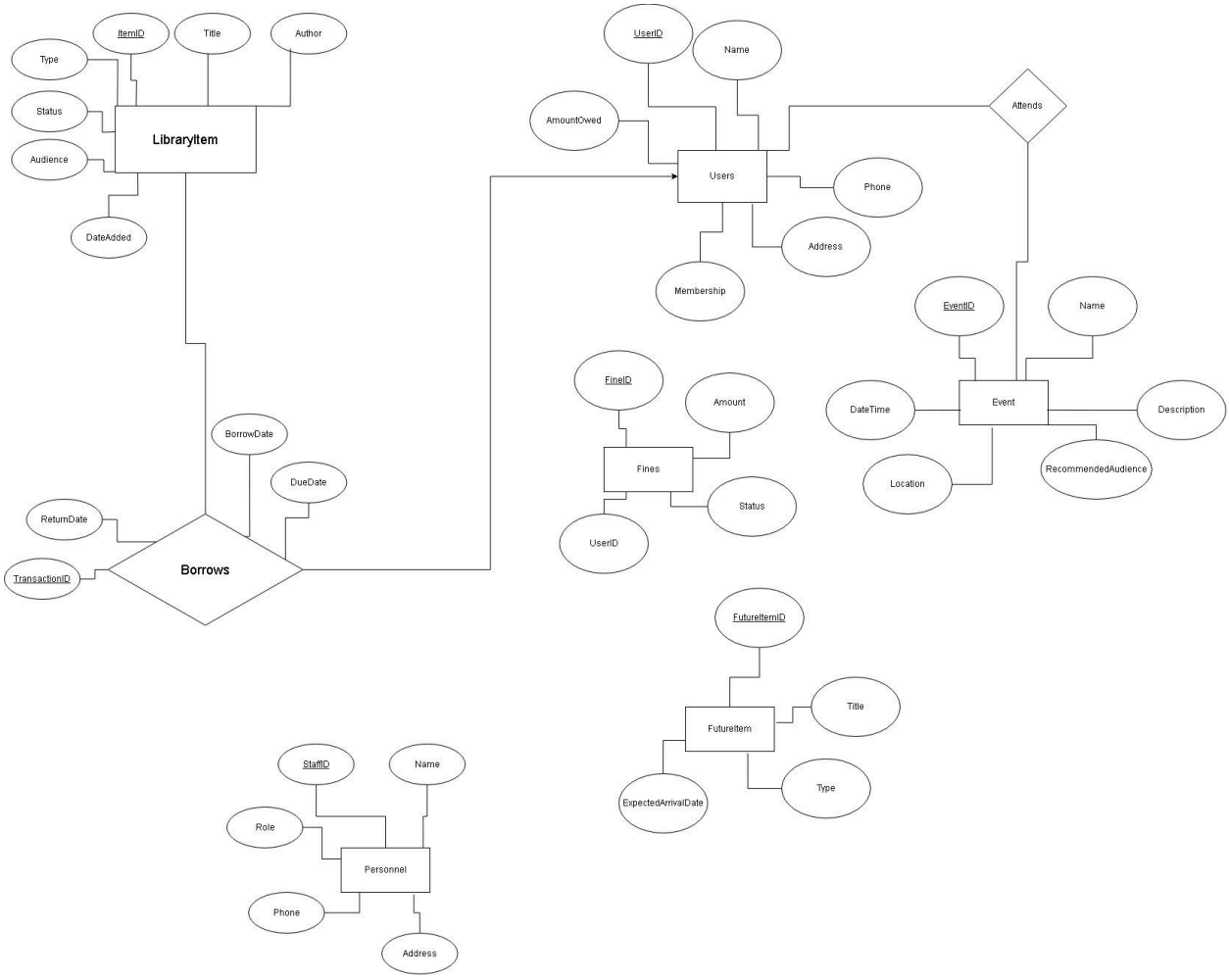
Personnel(StaffID, Name, Role, Phone, Address)
- **PK**: StaffID

FutureItem(FutureItemID, Title, Type, ExpectedArrivalDate)
- **PK**: FutureItemID

Attends(UserID, EventID)
- **FK**: UserID -> User(UserID) ***and*** EventID -> Event(EventID)

The following schemas will allow us to perform all necessary operations to run a library.
- The **LibraryItem** relation can store any item in the library that can be checked out. This generality is available due to ***type*** attribute.
- The **User** relation contains users who may check out items and also attend events. This works side-by-side with the **Borrows** relation to allow people to check out items in the library.
- The **Fine** relation is used to represent fines that need to be paid or have been paid.
- The **Event** relation is used to represent any events that may be held in the library. This works with **Attends** to record all library users who attend the events.
- The **Personnel** relation is used to keep record of all employees of the library.
- The **FutureItem** relation is used to record potential future items.

(3) E/R Diagrams



(4) Anomalies ?

Why anomalies are not allowed **:**

Redundancy Anomalies : All relations do not have redundancy anomalies because we have ensured that each relation is an ***abstraction*** of a single entity and that each attribute value of a tuple is unique to that tuple. Therefore there are no uncessary/unrelated attributes whose values are repeated in multiple tuples.

Update Anomalies : All relations do not have update anomalies because all necessary attributes of each entity are encapsulated into a single relation and no attributes which relate to the entity are specified in other relations (aside from identifiers/keys which will only be updated in extremely rare cases or never). So therefore, if an entity/tuple is being updated, we will not need to update any other relation.

Delete Anomalies : All relations do not have delete anomalies due to similar reasons for update anomalies. Each relation contains only the attributes necessary for a single tuple/entity, no extra information unrelated to the tuple/entity will be deleted.

Analyzing the functional dependencies **:**

User

- (non-trivial) **FD**s **:** (UserID)→(all attributes) , (Name, Phone, Address)→(all attributes)
- The FD involving the ID will almost always be used to determine users and do not intend to use other information to determine a user.
- If necessary, can use Name, Phone and Address attributes to determine all other attributes.
- Since all (non-trivial) FDs use a key (Name, Phone and Address is essentially a key) to determine information, the relation is in BCNF.

LibraryItem

- (non-trivial) **FD**s **:** (ItemID)→(all attributes) , (Title, Author, Type)→(all attributes)
- All (non-trivial) FDs use a key to determine the other attributes, therefore the relation is in BCNF.

Borrows

- (non-trivial) **FD**s **:** (TransactionID)→(all attributes)
- There are no more FDs, no other attributes which determine another attribute(s) in relation.
- The relation is in BCNF due to all FDs using a key to determine other attributes.

Fine

- (non-trivial) **FD**s **:** (FineID)→(all attributes)

- There are no more FDs, no other attributes which determine another attribute(s) in relation.
- The relation is in BCNF due to all FDs using a key to determine other attributes.

Event

- (non-trivial) **FD**s **:** (EventID)→(all attributes)
- There are no more FDs, no other attributes which determine another attribute(s) in relation.
- The relation is in BCNF due to all FDs using a key to determine other attributes.

Personnel

- (non-trivial) **FD**s **:** (StaffID)→(all attributes) , (Name,Role)→(all attributes)
- There are no more FDs, no other attributes which determine another attribute(s) in relation.
- The relation is in BCNF due to all FDs using a key to determine other attributes.

FutureItem

- (non-trivial) **FD**s **:** (FutureItemID)→(all attributes)
- There are no more FDs, no other attributes which determine another attribute(s) in relation.
- The relation is in BCNF due to all FDs using a key to determine other attributes.