Steps to make this diagram:

# Mapping of Strong Entity Types

The first step in the algorithm is to map the Strong Entity types. Create attributes in the relation for each attribute of the strong entity type which is atomic (single valued)

CREATED RELATIONS (TABLES):

BOOK (ISBN,Title,YearOfPublication)

* 1. Mapping the Author multivalued variable. Book has a mulivalued attribute of author. To Map this attribute create a database table. Create a Foreign Key(FK) to ISBN, the BOOK Primary Key.

AUTHOR(ISBN,AUTHORNAME)

PUBLISHER(PublisherName,Website,Country,State,City)  
MAGAZINE(ISSN,title)

COPY(COPYID,COPYLOSTFLAG,PURCHASEDATE,PRICE)

# Mapping of disjoint superclass/subclass

Reader entity can be one of faculty, student.

Mapping reader and creating a relation for the strong entity types reader, faculty, student

READER(READERID,NAME)

FACULTY(READERID)

* Attribute READERID is FK to READERID attribute in READER(READERID,NAME)

STUDENT(READERID)

* Attribute READERID is FK to READERID attribute in READER(READERID,NAME)

# Mapping of union type

Material relationship is a union of both Book and Magazine. To map such a union type, we generate a new Material\_ID for the Material relation and add this to both Book and Magazine as foreign keys.

# Mapping of relationships

**1:N relationships (N:1 relationships)**

For the mapping of 1:N or N:1 relations, we identify the relation representing the ‘N’ side. In this relation, we include the primary key of the relation representing the 1 side of the relationship.

In our figure, the ‘HAS’ relationship, between MATERIAL and COPY, the N-side is on the COPY relation (the 1…n notation is reversed). Thus, we include the Material\_ID field of the MATERIAL relation as a foreign key in the COPY relation.

**M: N relationships**

For the mapping of M: N or N: M relationships, we must create an entirely new relation for their representation. We include the primary keys of both the participating relations in the M: N relationship as the attributes of the new relation. Any attributes related to the relation itself are to be included as attributes of the new table.

**Mapping of ternary relationship**

For mapping of ternary relationships, we create a new relation and include in it, the primary keys of each of the participating entities.

In our example, Borrow is a ternary relationship. To map this, we create a new relation, and include, as its attributes, the Reader\_ID from READER, Copy\_ID from COPY and Checkout\_time from RECORD tables.

# Mapping of Weak Entity types

For the mapping of weak entity types, (e.g.) Magazine and Issues, we create a relation for the weak entity and include the primary key of the owner entity in the weak relation.

In this case, we include the ISSN field of the Magazine relation in the Issues relation as a foreign key, and keep the Vol\_Issue\_No field as the partial key.

Database Schema

Material (Material\_ID: String)

Book (Material\_ID: String, ISBN: Int, Title: String, Year of Publication: Date, Publisher\_Name: String)

Author (ISBN: Int, Author\_Name: String)

Publisher (Publisher\_Name: String, Website: String, Country: String, State: String, City: String)

Magazine (Material\_ID: String, ISSN: Int, Title: String, Publisher\_Name: String)

Copy (Copy\_ID: String, Price: Float, Purchase\_Date: Date, Copy\_Lost\_Flag: Boolean, Material\_ID: String)

Reader (Reader\_ID: String, Name: String)

Faculty (Reader\_ID: String)

Student (Reader\_ID: String)

Reservation (F\_Reader\_ID: String, Material\_ID: String, Reservation\_Date: Date, Close\_Date: Date, Exp\_Date: Date)

Record (Reader\_ID: String, Checkout\_Time: String, Return\_Date: Date, Due\_Date: Date)

Borrow (Reader\_ID: String, Copy\_ID: String, Checkout\_Time: String)

Fine (Reader\_ID: String, Checkout\_Time: String, Reason: String, Amount: Float, Issue\_Date: Date, Payement\_Date: Date)

Issues (ISSN: Int, Volume\_No: Int, Issue\_No: Int, Publication\_Date: Date)

Group Members:

Abhishek Bansal : axb146030

Aniket Jiddigoudar : anj140530

Stavan Patel : sxp147630