**CS631008 Data Mgt Systems Design**

**Project Deliverable 2: Logical Design**

**Team Members:**

Shiva Karthik Pinjarle Manmohan

Nithin Krishna Krishnappa

Tejas Belakavadi Kemparaju

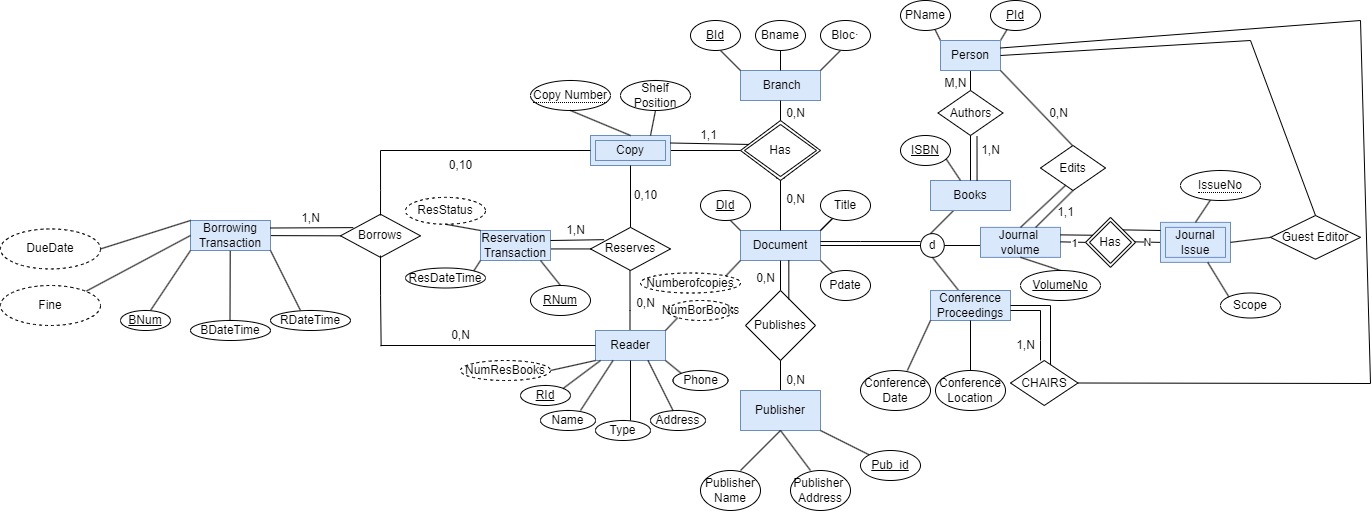
**1. Goals of this Phase:**

- Understand the requirements of the City Library system thoroughly.

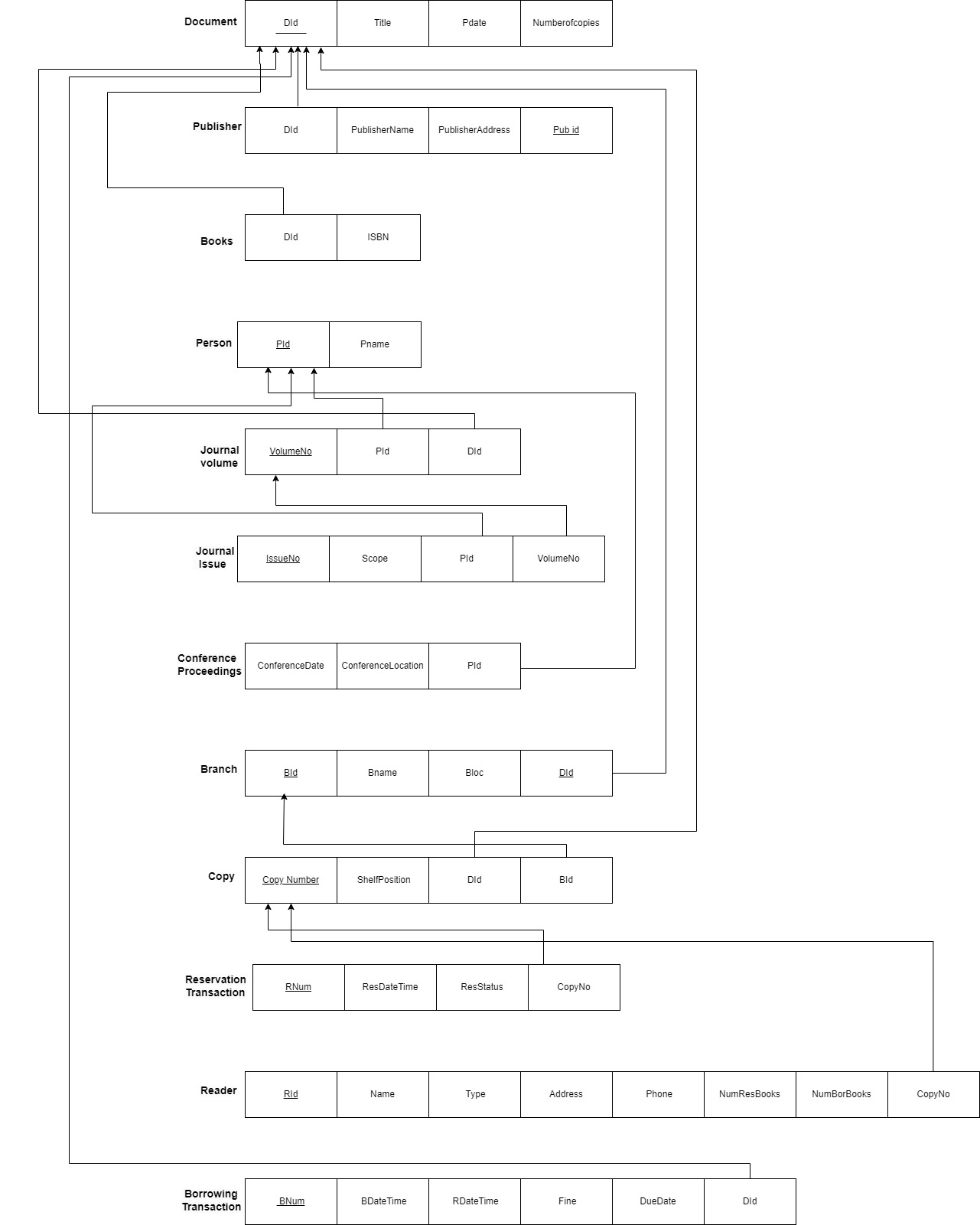
- Develop a logical database design by mapping the Extended Entity-Relationship (EER) schema to a Relational schema.

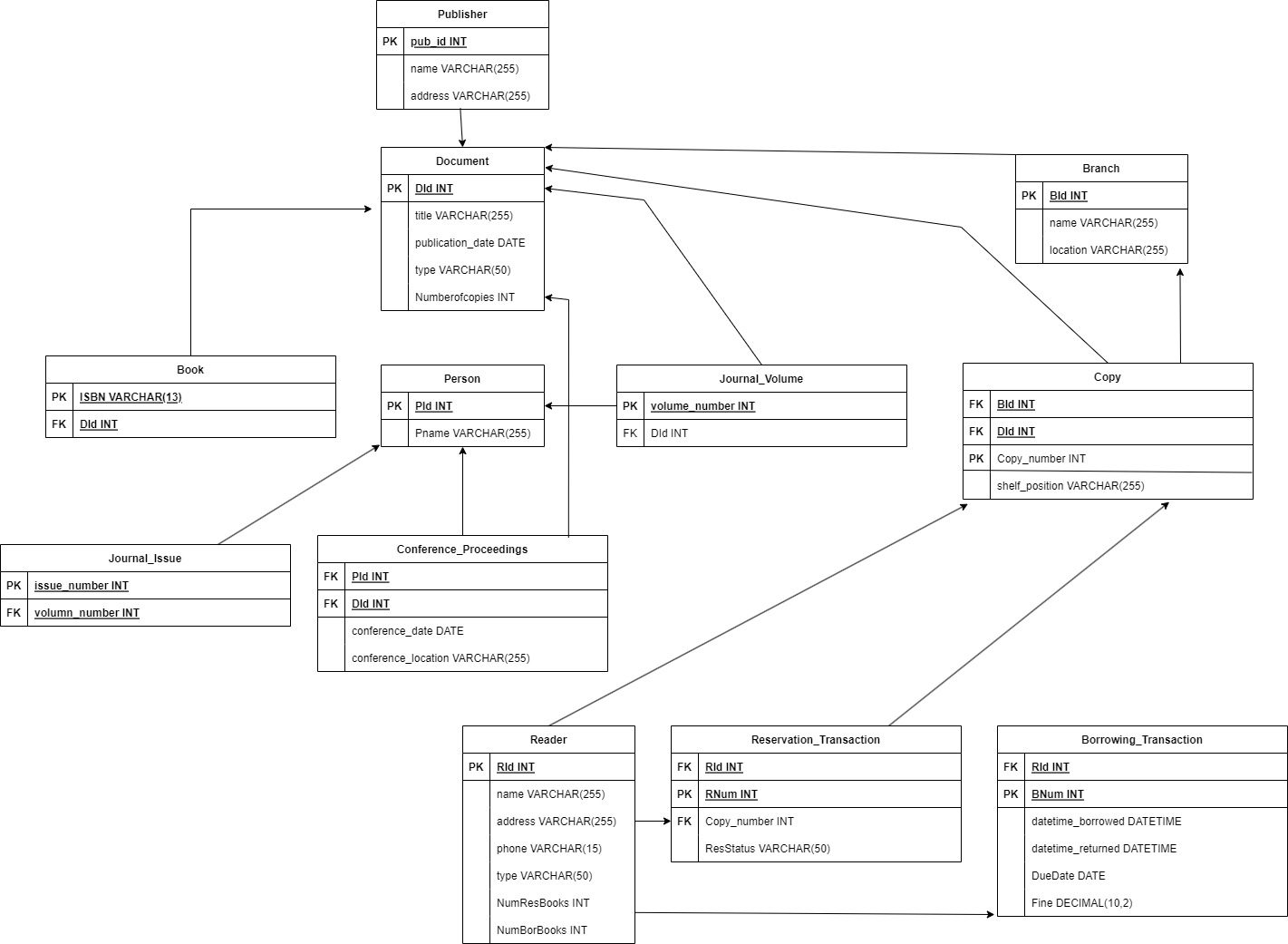
- Identify any necessary revisions to the previous phase's specifications and incorporate them into the database design.

**Modified Copy of ER**



**2. EER to Relational Mapping:**





**Constraints:**

1. Document Table:

- Each document must have at least one associated publisher.

- Each document copy must be associated with a specific branch.

- A document cannot have multiple instances with the same DId.

2. Publisher Table:

- Each publisher must have published at least one document.

3. Person Table:

- Every person must have a unique PId.

4. Branch Table:

- Every branch must exist in the system.

5. Copy Table:

- Each copy must be associated with a branch.

- A copy cannot have multiple instances with the same Copy\_Id.

6. Reader Table:

- Every reader must be registered in the system.

7. Reservation Table:

- Reserved copies must be picked up before 6 pm on the scheduled day, or the reservation is canceled.

- Each reservation transaction must have at least one document copy associated with it.

- A reader cannot reserve more than 10 documents simultaneously.

8. Borrowing Table:

- Each borrowing transaction must have at least one document copy associated with it.

- A reader cannot borrow more than 10 documents simultaneously.

**Logical Design**

-- Publisher Table

CREATE TABLE Publisher (

Pub\_Id INT PRIMARY KEY,

name VARCHAR(255),

address VARCHAR(255),

CONSTRAINT chk\_publisher\_document\_exists CHECK (

EXISTS (SELECT 1 FROM Document WHERE Pub\_Id = Publisher.Pub\_Id)

)

);

-- Person Table

CREATE TABLE Person (

PId INT PRIMARY KEY,

Pname VARCHAR(255),

CONSTRAINT uk\_person\_pid UNIQUE (PId)

);

-- Document Table

CREATE TABLE Document (

DId INT PRIMARY KEY,

title VARCHAR(255),

publication\_date DATE,

type VARCHAR(50),

Numberofcopies INT,

Pub\_Id INT,

CONSTRAINT fk\_document\_publisher FOREIGN KEY (Pub\_Id) REFERENCES Publisher(Pub\_Id),

CONSTRAINT chk\_document\_publisher\_exists CHECK (Pub\_Id IS NOT NULL),

CONSTRAINT uk\_document\_did UNIQUE (DId)

);

-- Book Table

CREATE TABLE Book (

DId INT PRIMARY KEY,

ISBN VARCHAR(50),

CONSTRAINT fk\_book\_document FOREIGN KEY (DId) REFERENCES Document(DId),

CONSTRAINT uk\_book\_isbn UNIQUE (ISBN)

);

-- Journal Volume Table

CREATE TABLE Journal\_Volume (

DId INT PRIMARY KEY,

volume\_number INT,

CONSTRAINT fk\_journal\_volume\_document FOREIGN KEY (DId) REFERENCES Document(DId)

);

-- Journal Issue Table

CREATE TABLE Journal\_Issue (

volume\_number INT,

issue\_number INT,

CONSTRAINT pk\_journal\_issue PRIMARY KEY (volume\_number, issue\_number),

CONSTRAINT fk\_journal\_issue\_journal\_volume FOREIGN KEY (volume\_number) REFERENCES Journal\_Volume(volume\_number)

);

-- Conference Proceedings Table

CREATE TABLE Conference\_Proceedings (

DId INT PRIMARY KEY,

conference\_date DATE,

conference\_location VARCHAR(255),

PId INT,

CONSTRAINT fk\_conference\_proceedings\_person FOREIGN KEY (PId) REFERENCES Person(PId),

CONSTRAINT fk\_conference\_proceedings\_document FOREIGN KEY (DId) REFERENCES Document(DId)

);

-- Branch Table

CREATE TABLE Branch (

BId INT PRIMARY KEY,

name VARCHAR(255),

location VARCHAR(255),

CONSTRAINT uk\_branch\_bid UNIQUE (BId)

);

-- Copy Table

CREATE TABLE Copy (

Copy\_Id INT PRIMARY KEY,

DId INT,

BId INT,

copy\_number INT,

shelf\_position VARCHAR(50),

CONSTRAINT fk\_copy\_document FOREIGN KEY (DId) REFERENCES Document(DId),

CONSTRAINT fk\_copy\_branch FOREIGN KEY (BId) REFERENCES Branch(BId),

CONSTRAINT uk\_copy\_copyid UNIQUE (Copy\_Id),

CONSTRAINT uk\_copy\_did\_bid\_copynumber UNIQUE (DId, BId, copy\_number)

);

-- Reader Table

CREATE TABLE Reader (

RId INT PRIMARY KEY,

name VARCHAR(255),

address VARCHAR(255),

phone VARCHAR(20),

type VARCHAR(50),

NumResBooks INT,

NumBorBooks INT,

CONSTRAINT uk\_reader\_rid UNIQUE (RId)

);

-- Reservation Transaction Table

CREATE TABLE Reservation\_Transaction (

RNum INT PRIMARY KEY,

RId INT,

ResStatus VARCHAR(20),

copy\_number INT,

CONSTRAINT fk\_reservation\_transaction\_reader FOREIGN KEY (RId) REFERENCES Reader(RId),

CONSTRAINT fk\_reservation\_transaction\_copy FOREIGN KEY (copy\_number) REFERENCES Copy(Copy\_Id),

CONSTRAINT chk\_reservation\_transaction\_copy\_count CHECK (

(SELECT COUNT(\*) FROM Reservation\_Transaction WHERE RId = Reservation\_Transaction.RId) <= 10

),

CONSTRAINT chk\_reservation\_transaction\_pickup\_time CHECK (

ResStatus = 'picked\_up' OR ResStatus = 'canceled' OR (ResStatus = 'reserved' AND pickup\_time <= '18:00')

)

);

-- Borrowing Transaction Table

CREATE TABLE Borrowing\_Transaction (

BNum INT PRIMARY KEY,

RId INT,

copy\_number INT,

datetime\_borrowed DATETIME,

datetime\_returned DATETIME,

DueDate DATE,

Fine DECIMAL(10, 2),

CONSTRAINT fk\_borrowing\_transaction\_reader FOREIGN KEY (RId) REFERENCES Reader(RId),

CONSTRAINT fk\_borrowing\_transaction\_copy FOREIGN KEY (copy\_number) REFERENCES Copy(Copy\_Id),

CONSTRAINT chk\_borrowing\_transaction\_copy\_count CHECK (

(SELECT COUNT(\*) FROM Borrowing\_Transaction WHERE RId = Borrowing\_Transaction.RId) <= 10

)

);

**3. Difficulties Encountered:**

- Balancing the level of detail in the schema with the need for simplicity and clarity.

- Defining relationships and constraints accurately based on the given requirements.