Final Project: Part 1

Jesse Roberts

Step 1 - Business Rules:

* A title may have one, or two ISBNs, an ISBN may only have one title
* An ISBN may be 10 digits
* A title may have one or many authors
* A title usually has one publisher, a publisher can have one or many books
* An author may have one or many books (authors with zero books are not relevant to this library)
* A title may have one or many pages, pages may only have one title
* A title can only belong to one category, a category can have many books
* A book can only be donated by one friend
* A title can only have been purchased for one price
* A title can have one or many subject matters, and a subject matter zero or many books
* A title/ISBN can only have one year, but a year many titles
* A friend may have zero or many books, a book can only be checked out by one friend at a time
* A book can only be borrowed (checked out, in, length of time) once at a time

Step 2 – Relational Schema



Diagram

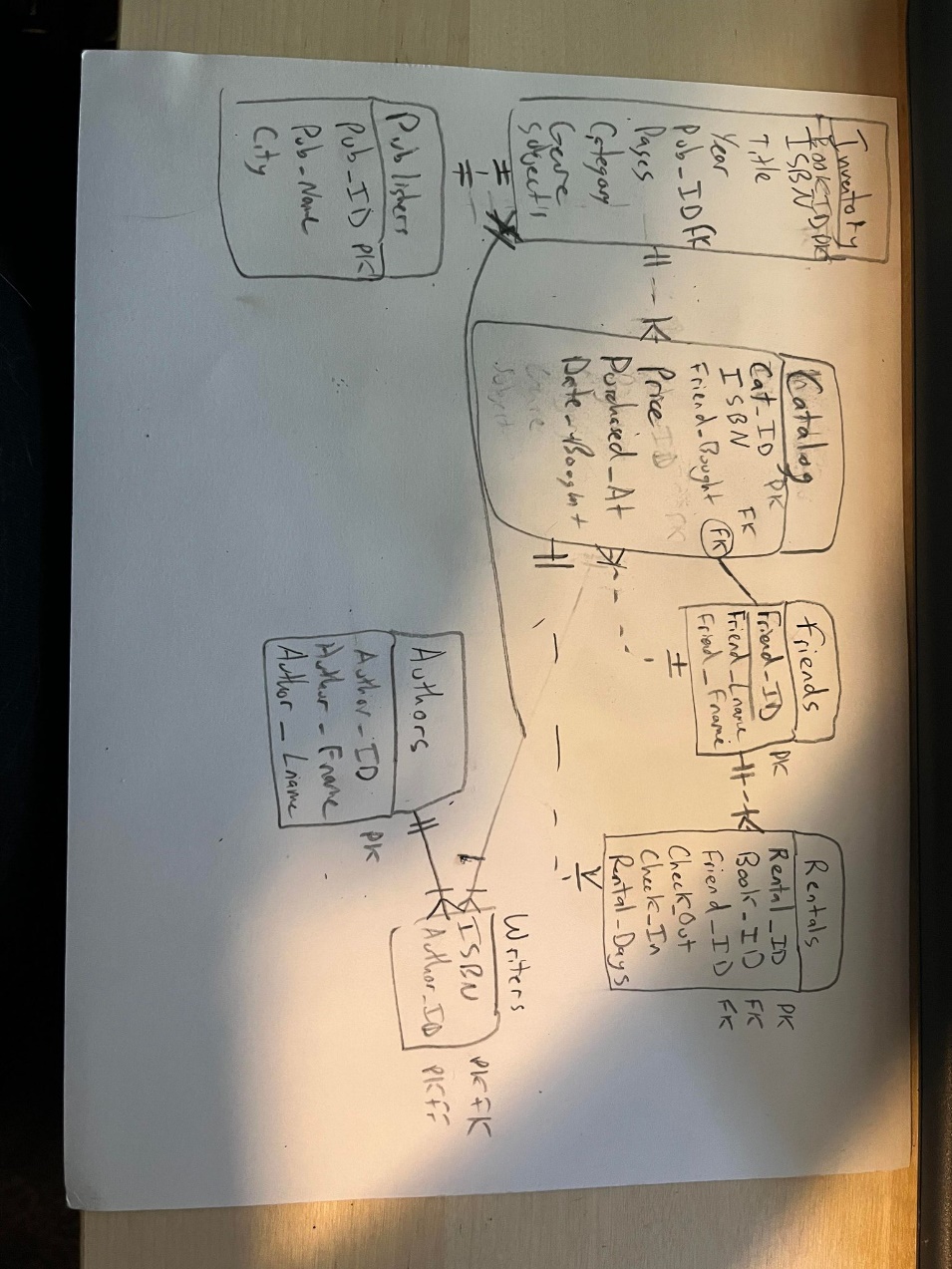
Description automatically generated with medium confidence

Diagram

Description automatically generated

Step 3 – Entity Relationship Model

I edited this for part two because I realized my friends table didn’t make sense, so this resembles your example more.



Step 4: Data Dictionary

A screenshot of a computer

Description automatically generated with low confidence

5.

Indexes:

1. ISBN (unique to each book, would work for catalog and inventory)
2. Friend\_ID (Need to easily query which friend belongs to which ID in case of rentals)
3. Book\_ID (unique to each copy of a book, in case a copy gets lost or its sticker comes off)
4. Rental\_ID (quickly discover what books were checked out in a rental
5. Author\_ID (quickly determine who wrote a specific book to seek out more books by same author)

Part 6 and **7 oops**:

CREATE TABLE `authors` (

  `Author\_ID` varchar(3) NOT NULL,

  `Author\_Fname` varchar(20) DEFAULT NULL,

  `Author\_Lname` varchar(20) DEFAULT NULL,

  PRIMARY KEY (`Author\_ID`),

  UNIQUE KEY `Author\_ID\_UNIQUE` (`Author\_ID`)

)

CREATE TABLE `catalog` (

  `Cat\_ID` int NOT NULL,

  `ISBN` char(10) NOT NULL,

  `Friend\_Bought` varchar(2) NOT NULL,

  `Price` double(5,2) DEFAULT NULL,

  `Purchased\_At` varchar(20) DEFAULT NULL,

  `Date\_Bought` date DEFAULT NULL,

  PRIMARY KEY (`Cat\_ID`),

  UNIQUE KEY `Cat\_ID\_UNIQUE` (`Cat\_ID`),

  UNIQUE KEY `ISBN\_UNIQUE` (`ISBN`),

  KEY `Friend\_Bought\_idx` (`Friend\_Bought`),

  CONSTRAINT `Friend\_Bought` FOREIGN KEY (`Friend\_Bought`) REFERENCES `friends` (`Friend\_ID`),

  CONSTRAINT `ISBN` FOREIGN KEY (`ISBN`) REFERENCES `inventory` (`ISBN`)

)

CREATE TABLE `friends` (

  `Friend\_ID` varchar(2) NOT NULL,

  `Friend\_Lname` varchar(20) NOT NULL,

  `Friend\_Fname` varchar(20) NOT NULL,

  PRIMARY KEY (`Friend\_ID`),

  UNIQUE KEY `Friend\_ID\_UNIQUE` (`Friend\_ID`)

)

CREATE TABLE `inventory` (

  `Book\_ID` varchar(4) NOT NULL,

  `ISBN` char(10) NOT NULL,

  `Title` varchar(30) NOT NULL,

  `Year` int DEFAULT NULL,

  `Pub\_ID` varchar(3) NOT NULL,

  `Pages` int NOT NULL,

  `Category` varchar(20) NOT NULL,

  `Genre` varchar(10) DEFAULT NULL,

  `Subject` varchar(15) DEFAULT NULL,

  PRIMARY KEY (`Book\_ID`),

  UNIQUE KEY `Book\_ID\_UNIQUE` (`Book\_ID`),

  UNIQUE KEY `ISBN\_UNIQUE` (`ISBN`),

  KEY `Pub\_ID\_idx` (`Pub\_ID`),

  CONSTRAINT `Pub\_ID` FOREIGN KEY (`Pub\_ID`) REFERENCES `publishers` (`Pub\_ID`)

)

CREATE TABLE `publishers` (

  `Pub\_ID` varchar(3) NOT NULL,

  `Pub\_Name` varchar(20) NOT NULL,

  `City` varchar(20) NOT NULL,

  PRIMARY KEY (`Pub\_ID`),

  UNIQUE KEY `Pub\_ID\_UNIQUE` (`Pub\_ID`)

)

CREATE TABLE `rentals` (

  `Rental\_ID` varchar(4) NOT NULL,

  `Book\_ID` varchar(4) NOT NULL,

  `Friend\_ID` varchar(2) NOT NULL,

  `Check\_Out` date DEFAULT NULL,

  `Check\_In` date DEFAULT NULL,

  `Rental\_Days` int DEFAULT NULL,

  PRIMARY KEY (`Rental\_ID`),

  UNIQUE KEY `Rental\_ID\_UNIQUE` (`Rental\_ID`),

  KEY `Book\_ID\_idx` (`Book\_ID`),

  KEY `Friend\_ID\_idx` (`Friend\_ID`),

  CONSTRAINT `Book\_ID` FOREIGN KEY (`Book\_ID`) REFERENCES `inventory` (`Book\_ID`),

  CONSTRAINT `Friend\_ID` FOREIGN KEY (`Friend\_ID`) REFERENCES `friends` (`Friend\_ID`)

)

CREATE TABLE `writers\_written` (

  `ISBN` char(10) NOT NULL,

  `Author\_ID` varchar(3) NOT NULL,

  PRIMARY KEY (`ISBN`,`Author\_ID`),

  KEY `Author\_ID\_idx` (`Author\_ID`),

  CONSTRAINT `Author\_ID` FOREIGN KEY (`Author\_ID`) REFERENCES `authors` (`Author\_ID`),

  CONSTRAINT `ISBN(10)` FOREIGN KEY (`ISBN`) REFERENCES `inventory` (`ISBN`)

)

Part 8:

INSERT INTO `jesse\_final`.`authors` (`Author\_ID`, `Author\_Fname`, `Author\_Lname`) VALUES ('0', 'Jerry', 'Springer');

INSERT INTO `jesse\_final`.`authors` (`Author\_ID`, `Author\_Fname`, `Author\_Lname`) VALUES ('1', 'Billy', 'Scapegoat');

INSERT INTO `jesse\_final`.`authors` (`Author\_ID`, `Author\_Fname`, `Author\_Lname`) VALUES ('2', 'Chicken', 'Cacchiatore');

INSERT INTO `jesse\_final`.`authors` (`Author\_ID`, `Author\_Fname`, `Author\_Lname`) VALUES ('3', 'Matt', 'Pandamiglio');

INSERT INTO `jesse\_final`.`authors` (`Author\_ID`, `Author\_Fname`, `Author\_Lname`) VALUES ('4', 'Mike', 'Birbiglia');

INSERT INTO `jesse\_final`.`friends` (`Friend\_ID`, `Friend\_Lname`, `Friend\_Fname`) VALUES ('0', 'Jampson', 'Johnny');

INSERT INTO `jesse\_final`.`friends` (`Friend\_ID`, `Friend\_Lname`, `Friend\_Fname`) VALUES ('1', 'Frejard', 'Bon');

INSERT INTO `jesse\_final`.`friends` (`Friend\_ID`, `Friend\_Lname`, `Friend\_Fname`) VALUES ('2', 'Aikikio', 'Sarasuka');

INSERT INTO `jesse\_final`.`friends` (`Friend\_ID`, `Friend\_Lname`, `Friend\_Fname`) VALUES ('3', 'Honch', 'Phillip');

INSERT INTO `jesse\_final`.`friends` (`Friend\_ID`, `Friend\_Lname`, `Friend\_Fname`) VALUES ('4', 'Flores', 'Corazon');

INSERT INTO `jesse\_final`.`publishers` (`Pub\_ID`, `Pub\_Name`, `City`) VALUES ('0', 'Randy Day Books', 'Norman, OK');

INSERT INTO `jesse\_final`.`publishers` (`Pub\_ID`, `Pub\_Name`, `City`) VALUES ('1', 'Rainy Day Books', 'OKC, OK');

INSERT INTO `jesse\_final`.`publishers` (`Pub\_ID`, `Pub\_Name`, `City`) VALUES ('2', 'Fancy Dance Books', 'Tulsa, OK');

INSERT INTO `jesse\_final`.`publishers` (`Pub\_ID`, `Pub\_Name`, `City`) VALUES ('3', 'Bookshelf Books', 'Not Real, OK');

INSERT INTO `jesse\_final`.`publishers` (`Pub\_ID`, `Pub\_Name`, `City`) VALUES ('4', 'Local Book Company', 'Faketown, OK');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('0', '1001100001', '\"The Berry the Boy Eats\"', '1999', '3', '234', 'Fiction', 'Mystery', 'Murder');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('1', '9999999999', '\"Jerry Spring: a Memoir\"', '2005', '2', '1902', 'Non-Fiction', 'Biography', 'Memoir');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('2', '1001190000', '\"Filth\"', '2014', '4', '1', 'Fiction', 'Fantasy', 'Assassins');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('3', '1001110111', '\"Insurrection\"', '2011', '4', '133', 'Reference', 'Encycl.', 'Jan 6');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('4', '1111111111', '\"Cowboys In Clothing\"', '2019', '1', '215', 'Fiction', 'Romance', 'Western');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('0', '1001100001', '\"The Berry the Boy Eats\"', '1999', '3', '234', 'Fiction', 'Mystery', 'Murder');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('1', '9999999999', '\"Jerry Spring: a Memoir\"', '2005', '2', '1902', 'Non-Fiction', 'Biography', 'Memoir');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('2', '1001190000', '\"Filth\"', '2014', '4', '1', 'Fiction', 'Fantasy', 'Assassins');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('3', '1001110111', '\"Insurrection\"', '2011', '4', '133', 'Reference', 'Encycl.', 'Jan 6');

INSERT INTO `jesse\_final`.`inventory` (`Book\_ID`, `ISBN`, `Title`, `Year`, `Pub\_ID`, `Pages`, `Category`, `Genre`, `Subject`) VALUES ('4', '1111111111', '\"Cowboys In Clothing\"', '2019', '1', '215', 'Fiction', 'Romance', 'Western');

INSERT INTO `jesse\_final`.`writers\_written` (`ISBN`, `Author\_ID`) VALUES ('1001100001', '1');

INSERT INTO `jesse\_final`.`writers\_written` (`ISBN`, `Author\_ID`) VALUES ('9999999999', '0');

INSERT INTO `jesse\_final`.`writers\_written` (`ISBN`, `Author\_ID`) VALUES ('1001190000', '3');

INSERT INTO `jesse\_final`.`writers\_written` (`ISBN`, `Author\_ID`) VALUES ('1001110111', '4');

INSERT INTO `jesse\_final`.`writers\_written` (`ISBN`, `Author\_ID`) VALUES ('1111111111', '2');

INSERT INTO `jesse\_final`.`rentals` (`Rental\_ID`, `Book\_ID`, `Friend\_ID`, `Check\_Out`, `Check\_In`) VALUES ('0', '1', '0', '1996-12-03', '1996-12-17');

INSERT INTO `jesse\_final`.`rentals` (`Rental\_ID`, `Book\_ID`, `Friend\_ID`, `Check\_Out`, `Check\_In`) VALUES ('1', '2', '0', '1997-12-01', '1997-12-27');

INSERT INTO `jesse\_final`.`rentals` (`Rental\_ID`, `Book\_ID`, `Friend\_ID`, `Check\_Out`, `Check\_In`) VALUES ('2', '1', '1', '1997-04-01', '1997-04-10');

INSERT INTO `jesse\_final`.`rentals` (`Rental\_ID`, `Book\_ID`, `Friend\_ID`, `Check\_Out`) VALUES ('3', '3', '2', '1998-03-25');

INSERT INTO `jesse\_final`.`rentals` (`Rental\_ID`, `Book\_ID`, `Friend\_ID`, `Check\_Out`, `Check\_In`) VALUES ('4', '4', '1', '2001-09-13', '2001-10-13');

**paRT 9:**

CREATE VIEW johnny\_rentals AS (

SELECT friends.Friend\_ID, Rental\_ID, Title, Friend\_Fname, Friend\_Lname, Friend\_Bought FROM catalog, inventory, friends, rentals

WHERE friends.Friend\_ID = 0 AND rentals.Friend\_ID = 0 AND Friend\_Bought = 0 AND inventory.ISBN = catalog.ISBN );**this view shows a specific friend’s rentals compared to his donated books.**

CREATE VIEW all\_friends AS (

SELECT DISTINCT a.Friend\_ID, Title, c.Book\_ID, Friend\_Fname, Friend\_Lname FROM friends a, rentals b, inventory c

WHERE a.Friend\_ID = b.Friend\_ID AND b.Book\_ID = c.Book\_ID);

**This view simply shows every rental, who checked it out, and what the book is called.**

**part 10:**

Conceptualizing

The beginning of any database design is creating business rules. These are the most abstract form of a database (no data yet exists, no tables, no fields). To create most of the business rules, I simply adapted the rules I was given for the assignment into usable constraints for database construction as we practiced in class. Just like in real life, the business rules are based on the simple facts of a given business. For example, a given book may have more than one author – because books simply can have multiple authors. A friend may be the purchaser of many books because a single person has the capacity to buy many books.

After establishing the business rules came the relational diagram. This one was especially difficult for me, and my classmates seemed to agree. The concept of normalization is confusing for those that understand tables and data from the start, I believe. Explaining the normalization process is easy, however, because each step has specific rules. For first normal form I made sure there were no duplicates and no possible entries with multiple answers, which at that point was all contained in my head. For second normal form, I removed partial dependencies. This in particular was difficult, because I didn’t really understand what a partial dependency was at first. Third normal form was much easier though, because removing transitive dependencies made sense.

After the relational schema, I drew an Entity Relationship Diagram. Actually, I drew multiple. My original was based on a flawed relational diagram and was missing important relationships. It also only included strong relationships, when weak relationships are necessary for this particular database. I refactored my initial drawing to more closely resemble the one you showed in class, but with the entities I wanted in my drawing to begin with.

The data dictionary was the easiest part, for me, but also very tedious. I simply made a table in Google Sheets containing every single field/attribute that could possibly exist, separated by table, and then marked which keys would be primary or foreign keys. This helped me visualize the database much more easily than the relational and entity relationship diagrams.

To start creating my database, I used the data dictionary to establish first what tables I wanted. Then I used CREATE SCHEMA and CREATE TABLE to establish them. In my CREATE TABLE statements, I specified the names of columns and what data types/constraints they would have. I utilized MySQL Workbench to assist in the constraint part. All of those columns, datatypes, and domains came from the data dictionary.

Then, I populated my field using a combination of INSERT statements and the Workbench GUI to create generated fields. Even though I created foreign keys in Workbench when I built the columns, the data did not auto populate and I’m not sure why.

To create my views I used CREATE VIEW viewname AS() and then selected the columns that made sense from each table, and specified that some columns have the same data so that no duplicates would be formed.

**Part 11:**

**Sql file attached**