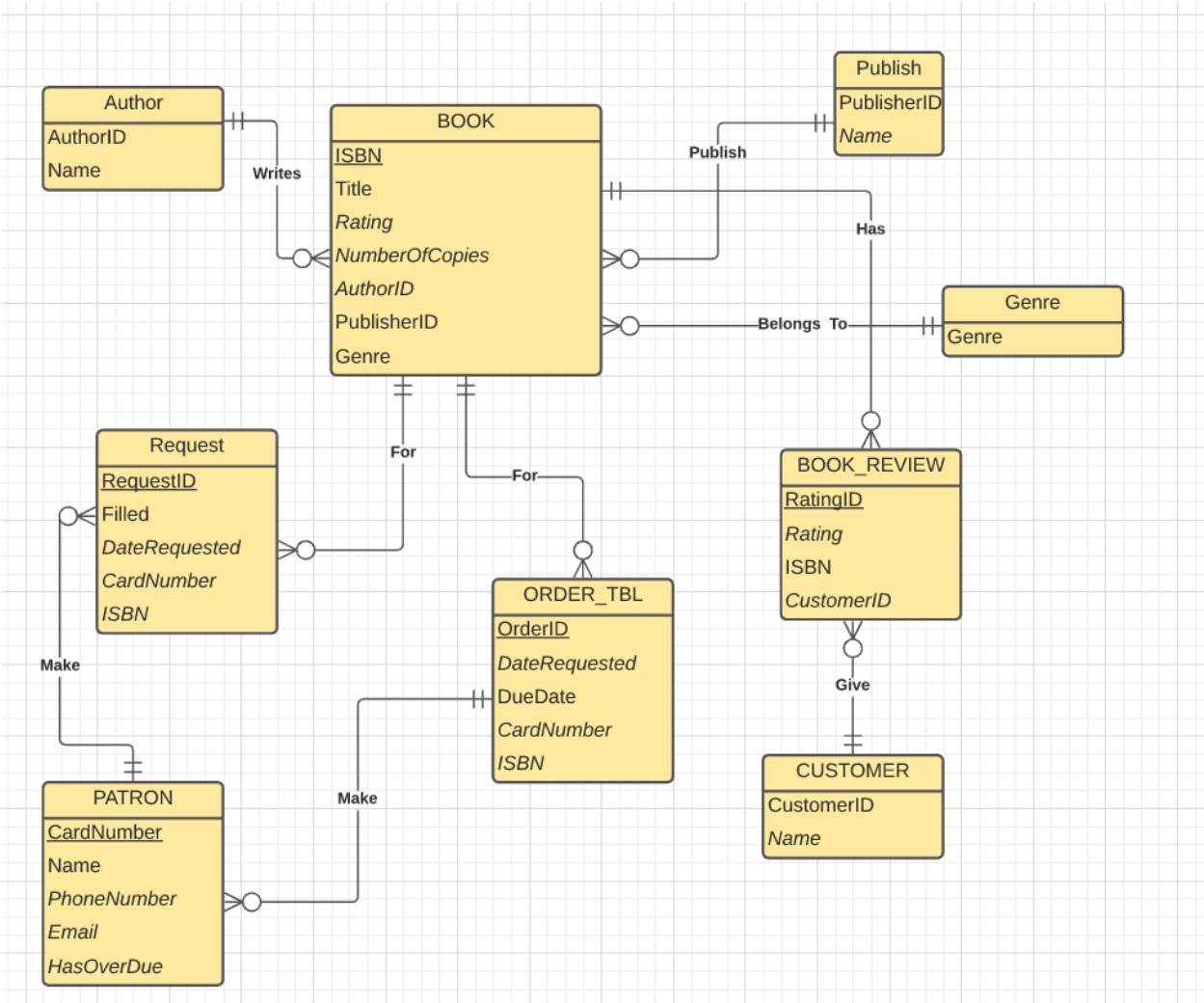
Group 2:
Daniel Whitlock
Abel Pinales
Saw Dennis Kham Myint Htar Oo
Murtadha Marzouq
Joshua Pucket

The ERD in Boyce-Codd Normal Form (BCNF) normalized form is (Description in the Appendix):



SCHEMA FORMAT V2 (NORMALIZED I USED TABLE NAMES instead of generic RELATIONS) EX:

R(A,B,C,D) { A -> BCD , BC -> AD , D -> B } --FUNCTIONAL DEPENDENCY

```
-- SCHEMAS in THE REQUESTED FORMAT
-- "RELATION_NAME(PrimaryKey, OtherColumn1, OtherColumn2,ForeignKey1)"


BOOK(ISBN , Title ,Genre Rating , Format ,NumberOfCopies ,Publisher , Author)

PATRON( CardNumber ,Name ,Active ,  PhoneNumber ,  Email ,  HasOverdue)

ORDER_TBL(OrderID ,DateRequested ,   CardNumber ,DueDate , PastDue ,   ISBN )


NEWBOOKREQUEST(NewBookISBN ,Title ,Genre ,Filled ,DateRequested , CardNumber)

PATRON_NEWBOOKREQUEST(Card_Number , NewBookISBN)


BOOK_REVIEW( RatingID ,  RATING ,  CustomerID ,  ISBN)


Blue = Foreign Key
RED = Primary Key
```

CREATE TABLES STATEMENTS:

http://34.150.201.142/team02/PROJECT/SQL/CREATE%20TABLES.TXT

INSERT DATA (2 working ways, SQL is easier than CSV):

A. We have SQL/TEXT file prepared:
   http://34.150.201.142/team02/PROJECT/SQL/
   http://34.150.201.142/team02/PROJECT/SQL/INSERT%20STATEMENTS.txt
B. We have CSV file prepared:
   http://34.150.201.142/team02/PROJECT/CSV/

IMPORT DATA (CSV) STATEMENTS:

```
--IN CASE YOU NEED THE CODE:
LOAD DATA INFILE '/location/of/file/tableName.csv'
INTO TABLE $whatevertableName is
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS; -- to not read the names, just the data
----------------------------------------
-- WE HAVE "INSERT STATEMENTS" ALSO TO MAKE IT EASIER.
```

IMPLEMENTED INSIDE THE DATABASE (team02):

APPENDIX:

Sharing the process:

TO RETHINK THE Schema to BCNF Normalized:

1. Book table

- The Book table is in unnormlaized form as One author can write more than one books and One Publisher can publish more than one book and More than one books are belongs to one genre. So we have to normalized it to make separate tables for Author, publisher and genre.

1. Book Table

Book (ISBN (PK), Title, Rating, Format, NumberOfCopies, Genre (FK), PublisherID (FK), AuthorID (FK))

2. Author Table

Author (AuthorID (PK), name)

3. Publisher Table

Publisher(PublisherID (PK) , name)

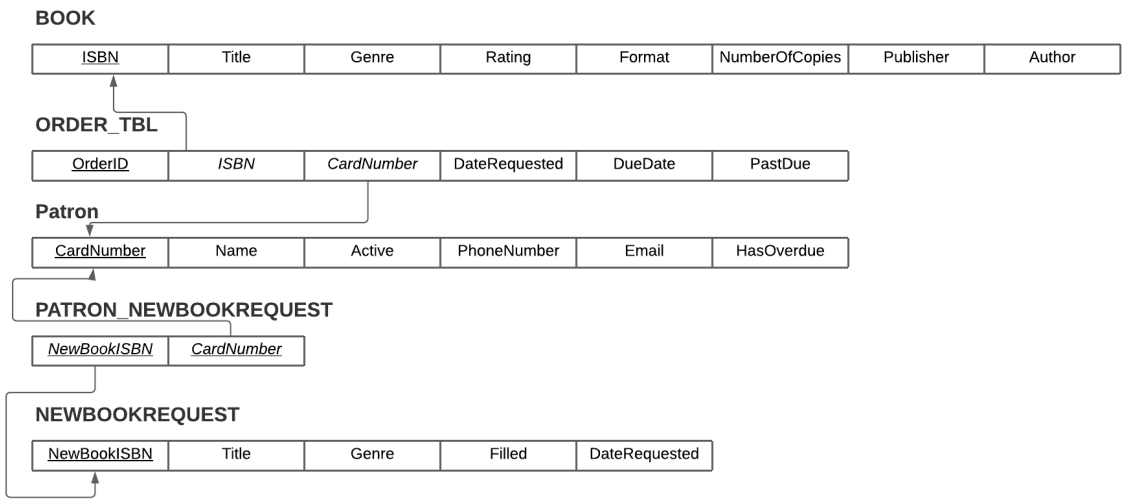
4. Genre Table

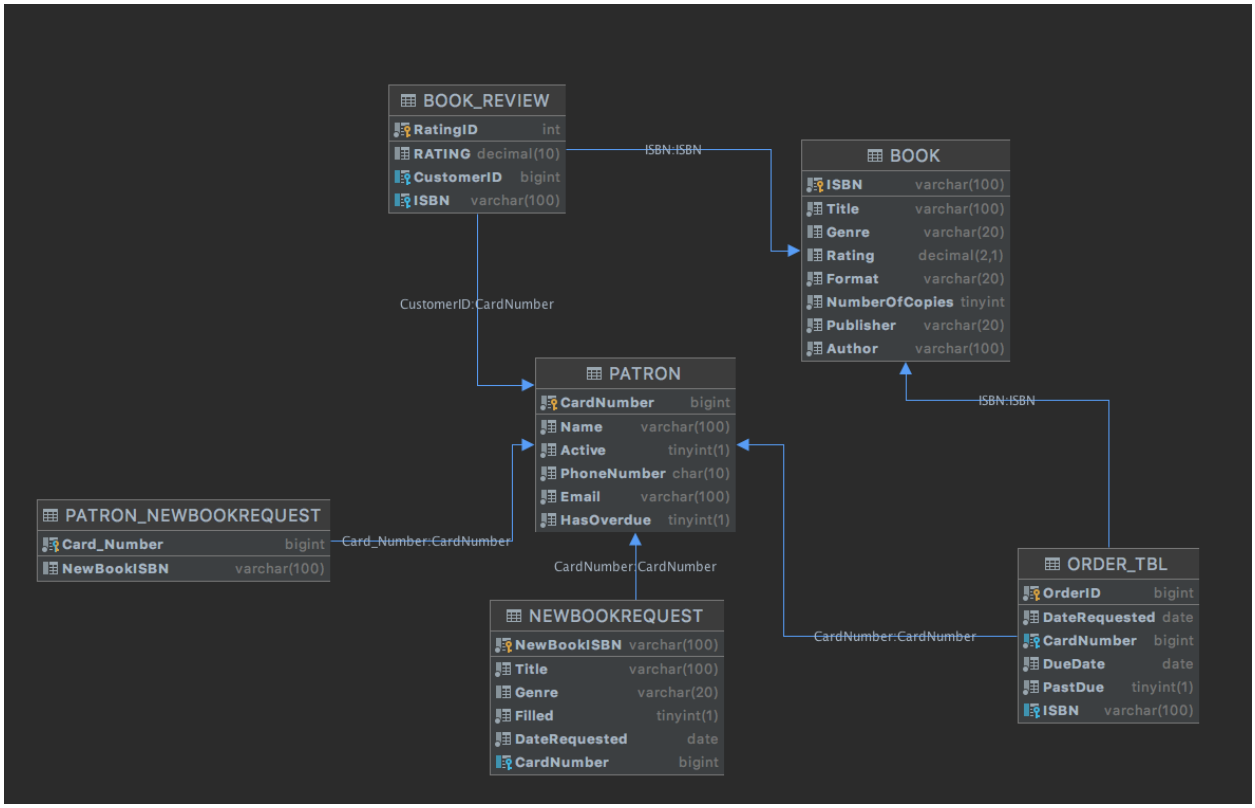
Genre (GenreID (PK)

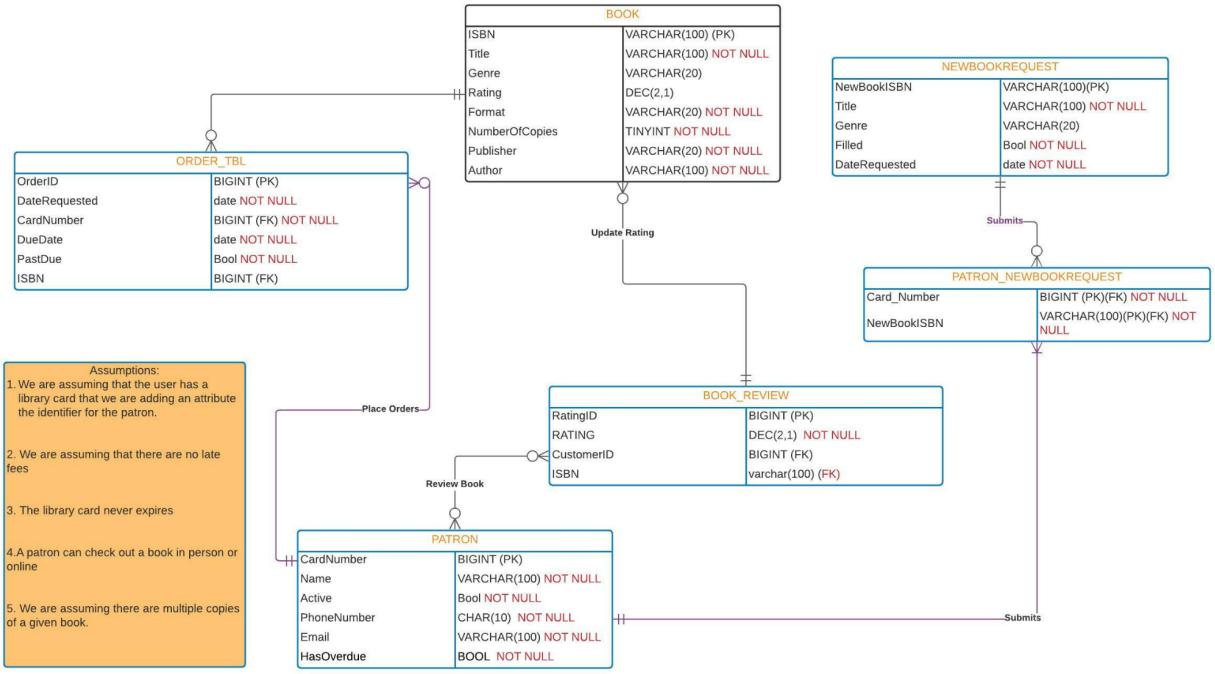
- 2. The table Patron_newNookRequest and NewBookRequest has duplicate attributes so we can replace this two tables with single request table and remove the duplicate attributes that are already present in Book table.

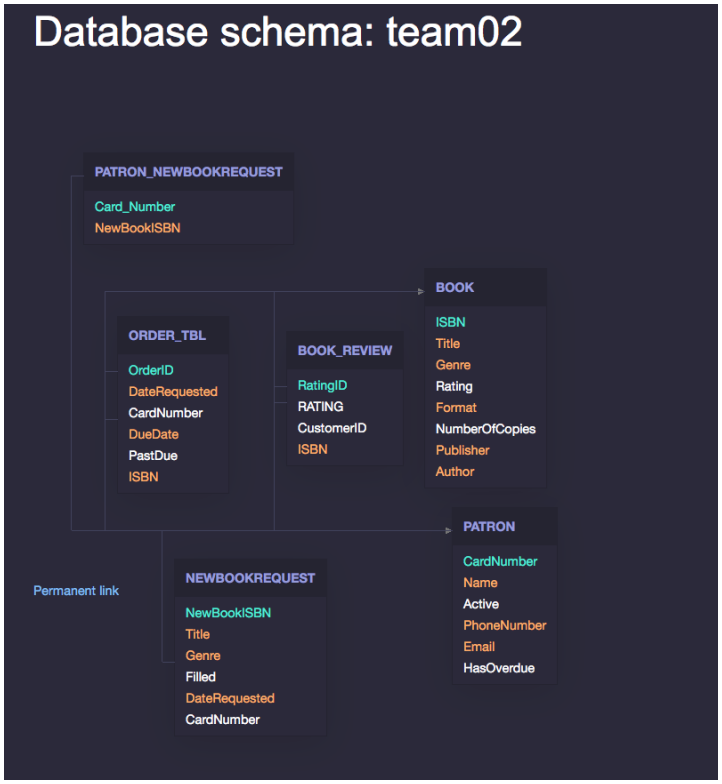
SCHEMA FORMAT V3:

NOTE: WE CREATED SEVERAL SCHEMA FORMATS TO COVER OUR BASIS



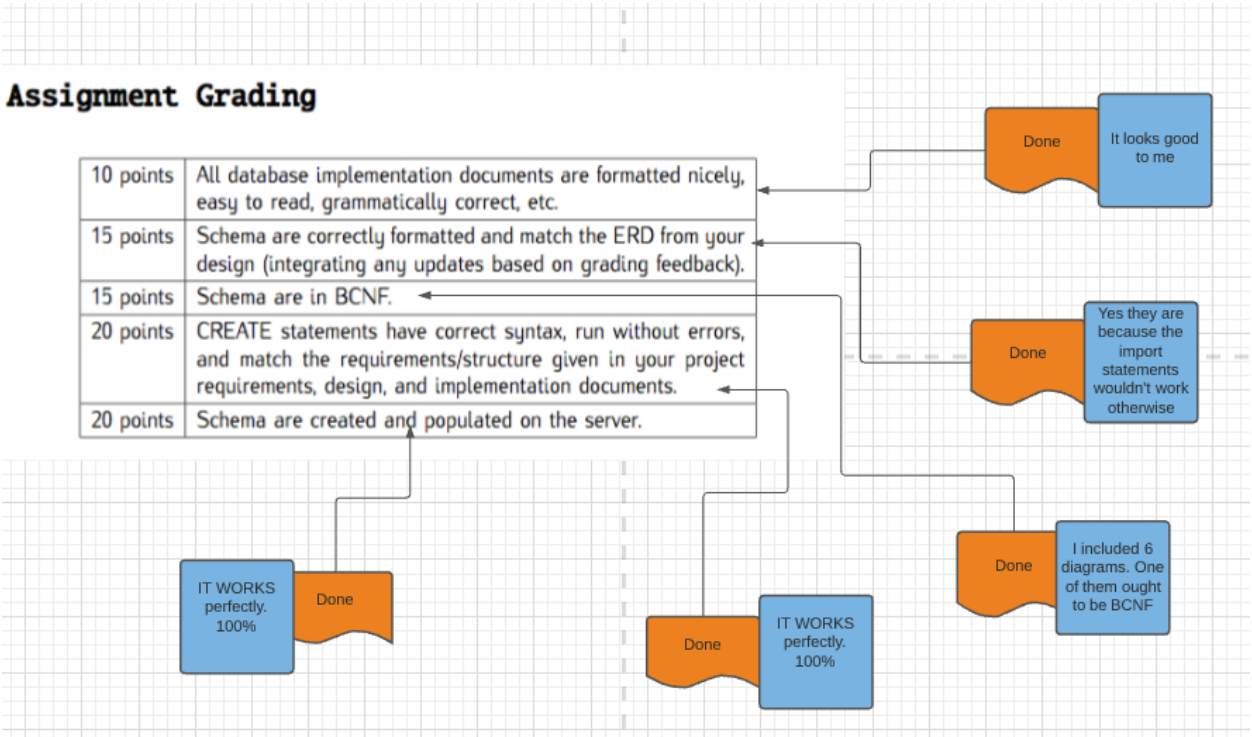

BOOK_REVIEW
RatingID int
RATING decimal(10)
CustomerID bigint
ISBN varchar(100)
ISBN:ISBN
BOOK
ISBN varchar(100)
Title varchar(100)
Genre varchar(20)
Rating decimal(2,1)
Format varchar(20)
NumberOfCopies tinyint
Publisher varchar(20)
Author varchar(100)
CustomerID:CardNumber
PATRON
CardNumber bigint
Name varchar(100)
Active tinyint(1)
PhoneNumber char(10)
Email varchar(100)
HasOverdue tinyint(1)
ISBN:ISBN
PATRON_NEWBOOKREQUEST
Card_Number bigint
NewBookISBN varchar(100)
Card_Number:CardNumber
CardNumber:CardNumber
NEWBOOKREQUEST
NewBookISBN varchar(100)
Title varchar(100)
Genre varchar(20)
Filled tinyint(1)
DateRequested date
CardNumber bigint
CardNumber:CardNumber
ORDER_TBL
OrderID bigint
DateRequested date
CardNumber bigint
DueDate date
PastDue tinyint(1)
ISBN varchar(100)


BOOK
ISBN VARCHAR(100) (PK)
Title VARCHAR(100) NOT NULL
Genre VARCHAR(20)
Rating DEC(2,1)
Format VARCHAR(20) NOT NULL
NumberOfCopies TINYINT NOT NULL
Publisher VARCHAR(20) NOT NULL
Author VARCHAR(100) NOT NULL
NEWBOOKREQUEST
NewBookISBN VARCHAR(100)(PK)
Title VARCHAR(100) NOT NULL
Genre VARCHAR(20)
Filled Bool NOT NULL
DateRequested date NOT NULL
ORDER_TBL
OrderID BIGINT (PK)
DateRequested date NOT NULL
CardNumber BIGINT (FK) NOT NULL
DueDate date NOT NULL
PastDue Bool NOT NULL
ISBN BIGINT (FK)
Update Rating
Submits
PATRON_NEWBOOKREQUEST
Card_Number BIGINT (PK)(FK) NOT NULL
NewBookISBN VARCHAR(100)(PK)(FK) NOT NULL
Assumptions:
1. We are assuming that the user has a library card that we are adding an attribute the identifier for the patron.
2. We are assuming that there are no late fees
3. The library card never expires
4.A patron can check out a book in person or online
5. We are assuming there are multiple copies of a given book.
Place Orders
BOOK_REVIEW
RatingID BIGINT (PK)
RATING DEC(2,1) NOT NULL
CustomerID BIGINT (FK)
ISBN varchar(100) (FK)
Review Book
PATRON
CardNumber BIGINT (PK)
Name VARCHAR(100) NOT NULL
Active Bool NOT NULL
PhoneNumber CHAR(10) NOT NULL
Email VARCHAR(100) NOT NULL
HasOverdue BOOL NOT NULL
Submits



NOTE: ALL DATA IS NOW OPERATIONAL ON THE **team02** DB.

Grading:



Bonus: (free DNS) the Apache server doesn't have a SSL Certificate, so HTTPS doesn't matter anyways. You will get an error either way:
http://itis3160.ddns.net/team02/PROJECT/