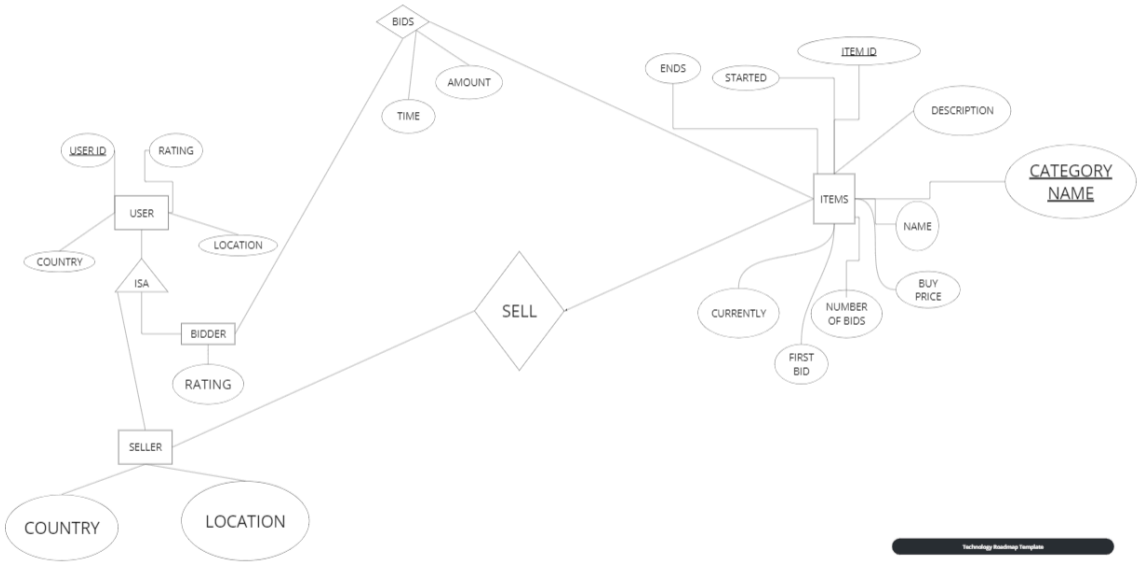
DESIGN OF ER MODEL AND RELATIONAL MODEL
IMPORTANT THINGS TO NOTE ARE JUST ER AND RELATIONAL MODEL-REST ARE JUST DESIGN NOTES



TABLES AND THEIR RELATIONAL SCHEMA

```
drop table if exists ITEM;

CREATE TABLE ITEM(

ITEMID NUMBERIC(11),

NAME CHAR(1024),

BUY_PRICE CHAR(100),

CURRENTLY CHAR(100),

FIRST_BID CHAR(100),

NUMBEROFBIDS NUMBERIC(11),

DESCRIPTION CHAR(10000),

STARTED CHAR(20),

ENDS CHAR(20),

CATEGORYNAME CHAR(1024) ,
```

```
PRIMARY KEY(ITEMID,CATEGORYNAME)

);

drop table if exists USER;

CREATE TABLE USER(

USERID CHAR(100),

RATING NUMERIC(11),

LOCATION CHAR(100),

COUNTRY CHAR(100),

PRIMARY KEY(USERID)

);

drop table if exists BIDDER;

CREATE TABLE BIDDER(

USERID CHAR(100),

RATING NUMERIC(11) NOT NULL,

PRIMARY KEY(USERID),

FOREIGN KEY (USERID) REFERENCES USER

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (RATING) REFERENCES USER

ON DELETE CASCADE

ON UPDATE CASCADE

);

drop table if exists SELLER;
```

```
CREATE TABLE SELLER(

USERID CHAR(100),

LOCATION CHAR(100) NOT NULL,

COUNTRY CHAR(100) NOT NULL,

PRIMARY KEY(USERID),

FOREIGN KEY (USERID) REFERENCES USER

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (LOCATION) REFERENCES USER

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (COUNTRY) REFERENCES USER

ON DELETE CASCADE

ON UPDATE CASCADE

);

drop table if exists SELL;

CREATE TABLE SELL(

ITEMID NUMERIC(11),

USERID CHAR(100),

CATEGORYNAME CHAR(1024),

PRIMARY KEY(ITEMID,CATEGORYNAME),

FOREIGN KEY (ITEMID) REFERENCES ITEM

ON DELETE CASCADE
```

```
ON UPDATE CASCADE,

FOREIGN KEY (CATEGORYNAME) REFERENCES ITEM

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (USERID) REFERENCES SELLER

ON DELETE CASCADE

ON UPDATE CASCADE

);

drop table if exists BIDS;

CREATE TABLE BIDS(

ITEMID NUMERIC(11),

TIME CHAR(20),

AMOUNT CHAR(20),

USERID CHAR(100) NOT NULL,

CATEGORYNAME CHAR(1024),

PRIMARY KEY(USERID,CATEGORYNAME,ITEMID),

FOREIGN KEY (ITEMID) REFERENCES ITEM

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (CATEGORYNAME) REFERENCES ITEM

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (USERID) REFERENCES BIDDER
```

```
ON DELETE CASCADE

ON UPDATE CASCADE

);
```

WORKING NOTES AND DETAILED ANALYSIS OF DESIGN PROCESS-

1) reason for using category and id as the unique key for an item-according to cow book and its chapter 3 exercises, there is no way to implement a total particpation of one entity set when it soesn't have a key constraint(we can model TP+key constraint for one entity set using rule 3 in chapter 3 of the cow book and can also model only key constraint but there is no way to model just TP constraint for one entity set in a relationship-but we need this as shown by the ER diagram for the items entity set-thus we make a new entity set category and use its primary key as a referencing key in the madeof relationship but to also account for the total participation constraint, we apply rule 3 of chapter 3 of cow book to change items entity set without creating an explicit relation called relation(in this modified items relation, we make both itemsid and category as a primary key and for TP constraint we make category field as not null-thus an item can now be there in more than one tuple and many categories can be there for one item and also because category is never null, we have both itemsid and category fields never null-thus every item has at least one category.

quoting from pragimtech website:"**Doesn't allow null values** and cannot contain duplicates. Values in an individual column can be duplicated, but across the columns they must be unique. Null values are not allowed in any columns in the composite primary key."-thus both itemsid and category cannot be null

2) a seller can seller many items and many a bidder can bid on many items

an item can be bid upon by many bidders but can only be sold by one seller

problem 1:we need a key constraint for items in a relationship with only the seller but not with the bidders, but the ER model contains a binary relationship with 2 role indicators which is equivalent to a pseudo ternary relationship. Thus if items gets an arrow in the bids relationship then it will be for both sellers and bidders but we don't want that.

If we try to use the second ER diagram where we don't have role indicators rather have ISA, there also we have to join both the sub relations seller and bidder to the items relation in a relationship called bid and then again this key constraint will be a problem.

SOLUTION: give 2 different relationships rather than one BID relationship for items-thus both the roles of users will be handling their relationship with items in a separate way-we have seen in the schema text file that seller has nothing to with bidding and thus we can make it have a different relationship called "SELL" with items while bidders can keep using bid relationship and in this scenario we have to use ISA to create 2 different entity sets called sellers and bidders

2 options for making attributes of child entity sets and the user entity set-

"We can map each of the entity sets Employees, Hourly Emps, and Contract Emps to a distinct relation. The Employees relation is created as in Section 2.2. We discuss Hourly Emps here; Contract Emps is handled similarly. The relation for Hourly Emps includes the hourly wages and hours worked attributes of Hourly Emps. It also contains the key attributes of the superclass (ssn, in this example), which serve as the primary key for Hourly Emps, as well as a foreign key referencing the superclass (Employees). For each Hourly Emps entity, the value of the name and lot attributes are stored in the corresponding row of the superclass (Employees). Note that if the superclass tuple is deleted, the delete must be cascaded to Hourly Emps. 2. Alternatively, we can create just two relations, corresponding to Hourly Emps and Contract Emps. The relation for Hourly Emps includes all the attributes of Hourly Emps as well as all the attributes of Employees (i.e., ssn, name, lot, hourly wages, hours worked)"

we also have the problem that a user can switch from one category/child entity set to another or be part of no child entity set

if we put all 4 attributes to the user then if we use 1st option, then we are making two child entity sets as exactly the same as the user entity set

if we put 2 attributes specific to the bidder and only id and rating to the user then how do we switch a user from being a seller to being a bidder(then he has to get 2 extra fields to become a bidder-note that a seller(while he is still a seller) can still have other 2 attributes and when he changes his state, we just transfer his tuple from seller to the bidder relation.

SOLUTION:we store all 4 attributes to user table and then just store its userid in both the child tables-so use case 1 for making 3 tables-child tables just contain one attribute

WE CAN ALSO USE 2+2 ATTRIBUTES CASE FOR USER AND SELLERS-when we transfer a user to the sellers relation, we first check if it has location and country-thus when we add an id to the sellers table,we check if it has location and country

when we use all 4 attributes in user table-

we can also use triggers and assertions-to state that if some userid is being inserted to seller table should have country and location because we don't have the location or country as part of

the sellers table-so we have to make a table constraint/trigger/assertion that we are taking these 2 attributes from the user table

when we use 2+2 case, we don't need triggers as we can use these 2 attributes directly from the seller table

problem: we can;t just transfer an sid from bidder to seller because in bidder we don't need location and country while in seller we need them-so we just can't store location and country of some users in seller while they reside in bidders because we don't have space for these 2 attributes for them in bidders

why choose these attributes for an item-

itemid cannot be empty

category cannot be empty

firstbid can be empty-thus currently can also be empty

buyprice can be empty

name can be empty

AN ITEM CAN HAVE NO SELLERS-THUS IF A SELLER IS DELETED WE CAN'T DELETE AN ITEM BUT IF A SELLER IS DELETED THEN ALL BIDS SHOULD BE DELETED AS NOW THERE IS NO POINT OF BIDDING(BUT STILL NO BIDDERS SHOULD BE DELETED)

THUS WE CAN'T USE 2ND APPROACH AS DESCRIBED IN CHAPTER 3 FOR MAKING KEY CONSTRAINTS IN SELL RELATIONSHIP-WE HAVE TO MAKE A NEW RELATION CALLED SELL AS WE CAN'T DELETE TUPLES OF WITHER OF THE PARTICIPATING RELATIONS AN THUS CAN'T USE ANY OF THE PARTICIPATING RELATIONS FOR INCORPORATING METHOD 2 OF CHAPTER 3 FOR KEY CONSTRAINTS

we are not putting userid as not null in seller relationship because an item is not supposed and necessarily be sold by a seller

tthus we need to store all 4 attributes in user table as a userid should be in bidders and sellers only if they are bidders and sellers

2 ways to go about-

1) 4 attributes in user table but not able to check if a tuple getting inserted into seller has location or country or not-how to maintain it in relational model(we can do in sql though by using sql query and transferring an id from user table to seller table)

2)2 attributes on seller but then we have redundancy of location and country attributes in the user and seller-if some user has all of attributes and is not yet a seller then we can't put it in any

of the child tables but then when it becomes a seller then it will have another copy of these 2 attributes in its fields/attributes

FINAL CALL-WE GO WITH WAY 2(WE MAKE SOME REDUNDANCY TO ENFORCE THE NOT NULL CONSTRAINT ON LOCATION AND COOUNTRY ATTRIBUTES FOR SELLERS-we prefer constraints over redundancy in relational model diagrams

//this may/may not work-use assertions

//this will work actually because these 2 tables inherit all attributes of the user table-so these 2 tables can use location and country attributes of the user table

WE CAN'T PUT DEFAULT FOR COUNTRY AND LOCATION IN THE USER TABLE AS SELLER CAN'T TAKE A DEFAULT VALUE OF NO COUNTRY FROM THE USER-NO COUNTRY MEANS THERE IS NO PROPER VALUE FOR THE SELLER AND THUS THIS USERID SHOULD NOT BE INCLUDED IN SELLER TABLE

#SAME-WE CAN'T PUT RATING AS DEFAULT VALUE IN USER AS BIDDER NEEDS A RATING

when country or location are deleted from the user table we also delete the user from the seller table because a seller cannot afford to lose its county and location

#every bidder is supposed to have a rating-so make this a new attribute of the seller table

NOTE:WE ARE MAKING THESE 3 ATTRIBUTES' REDUNDANCY ONLY BECAUSE WE ASSUME THAT THERE IS NO COVERING CONSTRAINT(THERE CAN BE SOME USERS WHO ARE BOTH NOT SELLERS AND BIDDERS)