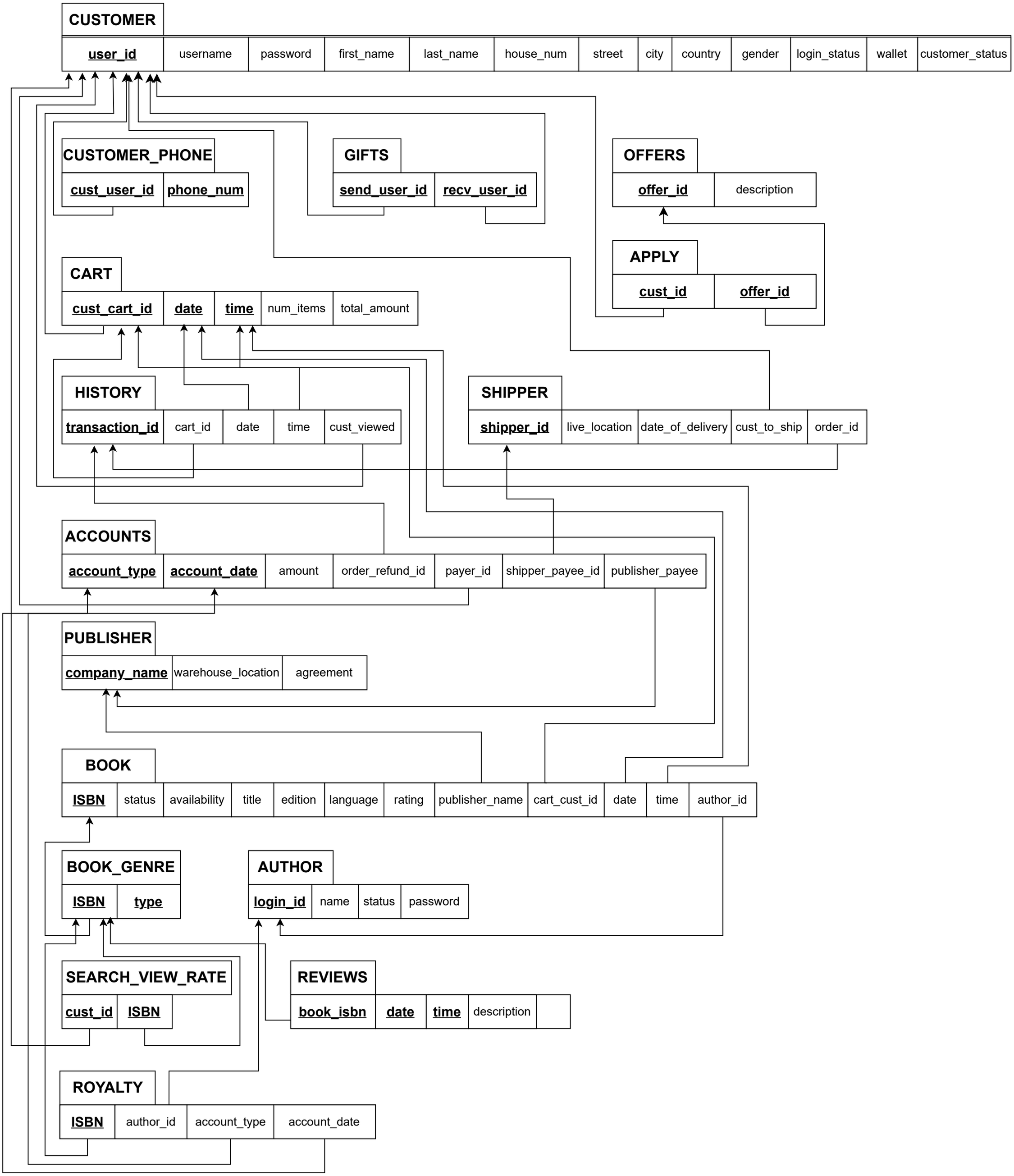
DNA PROJECT PHASE - 3

Aryan Mishra, Karthik Malavathula, Manas Mittal, Sushant Kumar
GROUP - 26

1.1 Realtional Schema in 1NF



1.2 1NF changes & justification

- composite attribute address for customers has been broken into simple attributes - house_num, street, city, country
- multi-valued attribute phone_number of customer removed and created as a separate entity with a foreign key constraint to customer's user_id and the primary key being union of the foreign key and a phone number of the customer
- multi-valued attribute types of the book removed and created as a separate entity with a foreign key constraint to the book's ISBN and the primary key being the union of the foreign key and a genre of the book
- Thus by resolving all composite and multi-valued attributes, our schema is in 1 normalized form
- the relationship sets have been reduced by either the foreign-key approach by adding a foreign key constraint on an attribute in the "many" side relation to the primary key of the "one" side or by cross-referencing in the case of many-to-many relations or 3-ary relationship "Royalty" where foreign key constraints exist for all participating entities but the primary key only includes the attributes referencing the "many" side relation(s).

1.2 2NF changes & justification

- NO CHANGES REQUIRED
- The schema is already in 1NF
- There exists no partial dependency - no non-prime attribute is dependent on any proper subset of any candidate key. In ALL of our relations, we uniquely determine the row using the primary key. Our primary is the minimal superkey and choice of such primary keys are in-line with the real world constraints of functional dependencies

1.2 3NF changes & justification

- NO CHANGES REQUIRED
- The schema is already in 2NF
- There is no transitive dependency of any non-prime attribute on any superkey. Note that we had a derived attribute "age" based on the customer's DOB. This could be thought of as a functional dependency where decomposition via normalization could be done. However, this would be an overkill and disperse the customer's information across 2 tables. So we decided to drop the derived entity "age" and instead capture it at the application level.