CS1555 Term Project: MyAuction

Group 4
Nicholas Amoscato [naa46@pitt.edu]
Ryan Sandhaus [rjs90@pitt.edu]

Limitations

- The top k highest root categories statistic was largely implemented in Java: a sorted map data structure stores the category and associated product counts. This implementation is acceptable based on Roxana's project clarifications; however, it is not ideal given the fact that a database could include many categories. An improvement could consist of abstracting this logic to a SQL function or procedure.

Implemented Improvements

- Fix buying_amount function to only include products that a bidder actually won (as opposed to incorporating all of the products the the bidder bid on).
- Wrote and tested queries for overlooked statistics including the top k highest volume leaf categories, the top k highest volume root categories, the top k most active bidders and the top k most active buyers.
- Limit the product_count function to only products that were sold.
- Update product statistics to include the highest bidder's login name (if the product is not sold) and the buyer's login name (if the product is sold).
- Move the category validity check from the SQL put_product procedure to Java in order to incorporate a better user experience and more descriptive warning messages that were not feasible with SQL exception handling in JDBC. The varray type used as input to the procedure (for multiple categories) was also changed to a table of varchar2's to conform with Oracle's JDBC ARRAY object.
- Alter product statistics to only include sold and under auction products (as opposed to sold and not sold products). Additionally, the query now includes products with no bids. This special case is handled in Java's output to prevent confusion.
- Fix suggestions to define a user's friend as one who has bid on *all* of the same products as the user (as opposed to our initial assumption which defined a user's friend as one who has bid on *some* of the same products as the user). This query was implemented with relational division in mind (implemented with the NOT EXISTS condition).
- Remove unnecessary amount2 attribute from the product table which kept track of the second highest bid on a particular product. This initial addition was implemented to simplify the query involved in selling products; however, as the development of this project continued, it was realized that updating this extra attribute after every bid was more costly than a relatively involved nested query when selling the product. The latter does not happen as often – not to mention removing the unnecessary attribute saves space.