Json Table:

**auction**:

uint64\_t ticket\_id; // primary key

uint64\_t game\_id; // secondary key

name auction\_owner;

uint64\_t highest\_bid;

name top\_bidder;

uint64\_t end\_date;

**ticket\_id:** Id of the auction. auction\_ID == ticket\_ID

**game\_id**: Id of the game for the ticket (not needed but allows us to have easy access to game for sorting)

**auction\_owner:** user who creates the auction. Will be the same as ticket\_owner so upon auction creation only a ticket\_id will be needed to supply this information.

**highest\_bid:** current top bid for the auction. The auction owner will specify the starting bid

**top\_bidder:** name of user who currently holds the highest bid on the auction. Will be assigned to name of **auction\_owner** if no one has bidded on the auction

**end\_date:** time when the auction is over. Will either be time when game tickets are handed out or a predetermined number of hours after the creation of the auction (whichever comes first)

**Functions probably needed:**

**createauction**:

Creates the auction

Params:

uint64\_t: ticket\_ID: id of the ticket

uint64\_t: Starting\_bid: value to begin bidding at in 100th of HTK (ex 5.00 HTK == 500)

uint64\_t: Auction end date in YYYYMMDDhhmm

Fails if the ticket is a lottery ticket

Fails if the starting bid is lower than the face value of the ticket

This action will need to be a new contract in hokipoki.cpp

**bid**:

Places a bid on the auction

Params:

uint64: auction\_ID: id of auction

string user: user applying the bid

uint64 bid: amount user is bidding

Notes:

If highest\_bid (auction) < bid and balance(user) >= bid, highest\_bid (auction) HTK are transferred from hokipoki back to top\_bidder (auction), bid HTK are transferred from user to hokipoki, highest\_bid is set to bid, and top\_bidder is set to user. If top\_bidder (auction) == auction\_owner (auction) then only bid HTK are transferred from user to hokipoki and highest\_bid (auction) and top\_bidder (auction) are set accordingly.

If highest\_bid (auction) >= bid. Nothing happens (maybe user is told to bid higher xd)

(tentative) bid must be at least 1.00 HTK higher than the current bid

This action will need to be a new contract in hokipoki.cpp

**execauction**:

Called after the end date of the auction, by either the owner of the ticket or the winner of the auction.

Params:

uint64 auction\_ID: id of auction

Notes:

Transfers ticket to top\_bidder (auction) and transfers highest\_bid (auction) HTK to owner (auction). If no one bidded on the auction, nothing happens.

This action will need to be TWO new actions in hokipoki.cpp

**execallauctns:**

Admin only  
 Executes all auctions that have passed their end time for a given game\_id

Params:

uint64 game\_id: game\_id to execute all lotteries

Notes:

Just a wrapper for Execute\_auction

Can be done entirely in libgoblin.py

**cancelauction**:

Required to be run by the owner of the auction

Params:

uint64\_t ticket\_id

Notes:  
 If there are no bids on the auction, deletes the auction

Please Add/Remove things that you think would improve it