**R4**

void Game::addItemToPlayer(const std::string& recipient, uint16\_t itemId)

{

Player\* player = g\_game.getPlayerByName(recipient);

if (!player) {

player = new Player(nullptr);

if (!IOLoginData::loadPlayerByName(player, recipient)) {

delete player; // Free memory if loading player fails

return;

}

}

Item\* item = Item::CreateItem(itemId);

if (!item) {

delete player; // Free memory if creating item fails

return;

}

g\_game.internalAddItem(player->getInbox(), item, INDEX\_WHEREEVER, FLAG\_NOLIMIT);

if (player->isOffline()) {

IOLoginData::savePlayer(player);

} else {

delete player; // Free memory if player is not offline

//The memory for the player is freed if the player is online, because there is no need to keep the player object after the item is added to their inbox.

}

}

**R3**

function removePlayerFromParty(targetPlayerId, memberName)

local targetPlayer = Player(targetPlayerId)

// Check if the target player exists

if not targetPlayer then

print("Target player does not exist")

return

end

local party = targetPlayer:getParty()

// Check if the player is in a party

if not party then

print("Target player is not in a party")

return

end

local partyMembers = party:getMembers()

local memberToRemove = Player(memberName)

for i, member in ipairs(partyMembers) do

if member == memberToRemove then

party:removeMember(member)

return

end

end

end

**R2**

function printSmallGuildNames(memberCount)

local selectGuildQuery = "SELECT name FROM guilds WHERE max\_members < %d;"

local resultId = db.storeQuery(string.format(selectGuildQuery, memberCount))

// Check if there was any error

if not resultId then

print("Error executing database query")

return

end

// we need a loop to print guild names because could be more than one

repeat

local guildName = result.getString("name")

print(guildName)

until not result.next(resultId)

resultId.free(resultId) //Free the result set

end

**R1**

local function releaseStorage(player)

if not player then

print("Error: Invalid player object")

return

end

//I'm assuming that setStorageValue returns a value to check if it was successful or not

local success = player:setStorageValue(1000, -1)

if not success then

print("Error: Failed to release storage for player")

end

end

function onLogout(player)

if not player then

print("Error: Invalid player object")

return false

end

if player:getStorageValue(1000) == 1 then

addEvent(releaseStorage, 1000, player)

end

return true

end