数据结构设计

游戏向战绩服务 写入数据 ganme -> record

message GamePlayerInfo {  
 uint64 UserId = 1; //用户Id  
 string Name = 2; //姓名  
 int32 BringinIntegral = 3; //带入积分  
 int32 WinIntegral = 4; //输赢积分  
}  
  
//games -> center  
//games -> center  
message WriteGameRecordReq {  
 uint64 RoomRecordId = 1; //游戏房间创建时生成的唯一Id 用户关联该房间  
 string GameId = 2; //游戏Id  
 int32 RoomType = 3; //房间类型 1 俱乐部房间 2 好友房间  
 int64 ClubId = 4; //俱乐部Id 当 AreaType 1 时需要填写  
 uint32 RoomId = 5; //房间号  
 int32 Index = 6; //游戏当前局数  
 int64 GameStartTime = 7; //开始时间 存储时间错  
 int64 GameEndTime = 8; //结束时间 存储时间错  
 repeated GamePlayerInfo PlayerInfos = 9; //游戏内玩家数据  
 repeated bytes RePlayData = 10; //复盘数据 暂时可不写  
}

Record 数据表设计

Usergamerecord 用户战绩表

//用户游戏记录  
type UserGameRecord struct {  
 UserId uint64 //用户Id  
 GameId string //游戏Id  
 RoomType int32 //房间类型 1俱乐部房间 2好友房  
 RoomId uint32 //房间号  
 Index int32 //第几局  
 GameStartTime time.Time //开始时间 存储时间错  
 GameEndTime time.Time //结束时间 存储时间错  
 WinIntegral int32 //输赢积分  
 RoomRecord \*mgo.DBRef //房间记录关联Id  
 GameRecord \*mgo.DBRef //游戏记录关联Id  
}

roomrecord

//房间记录  
type RoomRecord struct {  
 RoomRecordId uint64 //房间记录Id 关联房间的数据库唯一id  
 RoomId uint32 //房间号  
 GameId string //游戏Id  
 RoomType int32 //房间类型 1俱乐部房间 2好友房  
 ClubId int64 //俱乐部Id 当 AreaType 1 时需要填写  
 GamePlays []\*RoomPlayerInfo //游戏参与玩家信息 为了减少查询量 将必要信息进行存储  
 GameRecords []\*mgo.DBRef //房间内游戏数据 数组  
}

type RoomPlayerInfo struct {  
 UserId uint64 //用户Id  
 Name string //姓名  
 WinIntegral int32 //输赢积分 房间内的统计数据  
}

gamerecord

//游戏记录  
type GameRecord struct {  
 RecordId uint64 //游戏记录ID 主键  
 GameId string //游戏Id  
 AreaType int32 //场类型 1俱乐部房间 2好友房  
 ClubId int64 //俱乐部Id 当 AreaType 1 时需要填写  
 RoomId uint32 //房间号  
 Index int32 //第几局  
 GameStartTime time.Time //开始时间 存储时间错  
 GameEndTime time.Time //结束时间 存储时间错  
 GamePlays []\*GamePlayerInfo //游戏参与玩家信息 为了减少查询量 将必要信息进行存储  
}

Gamereplay

//游戏复盘记录 由于复盘数据过大type GameRePlayData struct {  
 RecordId uint64 //游戏记录ID  
 RePlayData []byte //复盘协议数据  
}

俱乐部统计

clubstatistics

//俱乐部统计数据 现在只统计7天内的  
type ClubStatisticsData struct {  
 ClubId int64 //俱乐部Id 当 AreaType 1 时需要填写  
 StatisticsTime time.Time //统计时间 到天 2019,20,26  
 IntegralStatistics []\*StatisticsData //积分统计  
 PlayStatistics []\*StatisticsData //对局次数统计 只在俱乐部中的游戏次数  
}

type StatisticsData struct {  
 UserId uint64 //用户Id  
 Statistics int64 //统计数据 积分统计 为积分 对局次数统计 为对局数  
}

当天的统计数据 缓存在Redis 数据中 每天固定时间 清除 并写入到俱乐部统计表中

//俱乐部 当天统计缓存数据  
type ClubStatisticsRedisData struct {  
 ClubStatistics map[int64]\*ClubStatisticsRedisItemData  
}  
  
type ClubStatisticsRedisItemData struct {  
 UserId uint64 //用户Id  
 StatisticsIntegral int64 //当天输赢积分统计  
 StatisticsPlay int64 //当天本俱乐部次数统计  
}