英雄联盟数据库实验报告

1. 系统需求分析
   1. 需求概述和系统边界

英雄联盟数据库旨在建立一个完整的数据库系统，以跟踪英雄联盟游戏中的各种数据和信息。该系统的边界涵盖了游戏中的各种实体，包括英雄、物品、符文、比赛、玩家、团队、皮肤以及匹配系统和排名等。

* 1. 主要业务处理流程

主要业务处理流程包括：

记录和管理每场比赛的相关信息，包括比赛的时间、持续时间、结果等。

追踪玩家的游戏表现，如击杀数、死亡数、助攻数等。

维护游戏中所有可用物品和符文的目录，包括它们的属性和效果。

跟踪玩家拥有的英雄、皮肤以及相关的进度和成就。

记录团队的形成和成员信息。

存储匹配系统和排名相关的数据，以便促进匹配制度的运作并跟踪玩家的排名进度。

* 1. 功能需求分析

功能需求是对系统所需功能的详细描述，包括用户所需的功能以及系统应该如何响应用户的需求。在英雄联盟数据库中，功能需求可以细分如下：

* + 1. 记录比赛详情

系统应能够记录每场比赛的相关信息，包括比赛时间、持续时间、比赛结果等。

系统应支持对比赛数据进行增加、删除、修改和查询操作。

系统应能够提供比赛数据的统计分析功能，如平均比赛持续时间、胜率等。

* + 1. 追踪玩家游戏表现

系统应能够追踪玩家在游戏中的表现数据，如击杀数、死亡数、助攻数、金币数、小兵击杀数等。

系统应支持对玩家表现数据进行增加、删除、修改和查询操作。

系统应能够提供玩家表现数据的统计分析功能，如击杀数、死亡数、助攻数的平均值、最大值、最小值等。

* + 1. 维护游戏元素信息

系统应能够维护游戏中所有可用物品和符文的目录，包括它们的属性和效果。

系统应支持对物品和符文信息进行增加、删除、修改和查询操作。

系统应能够提供物品和符文信息的统计分析功能，如物品和符文的数量、价格分布等。

* + 1. 跟踪玩家拥有的英雄和皮肤

系统应能够记录玩家拥有的英雄和皮肤信息，包括英雄的精通等级、皮肤的解锁状态等。

系统应支持对玩家拥有的英雄和皮肤信息进行增加、删除、修改和查询操作。

系统应能够提供玩家拥有的英雄和皮肤信息的统计分析功能，如玩家拥有的英雄和皮肤数量、解锁状态分布等。

* + 1. 记录团队的形成和成员信息

系统应能够记录团队的形成信息，包括团队ID、成员ID、角色等。

系统应支持对团队信息进行增加、删除、修改和查询操作。

系统应能够提供团队信息的统计分析功能，如团队成员数量、角色分布等。

* + 1. 存储匹配系统和排名相关数据

系统应能够存储匹配系统和排名相关的数据，包括匹配ID、玩家ID、匹配历史、排名比赛结果等。

系统应支持对匹配系统和排名数据进行增加、删除、修改和查询操作。

系统应能够提供匹配系统和排名数据的统计分析功能，如排名分数分布、匹配历史统计等。

* 1. 数据需求分析

数据需求分析是对系统中需要存储的数据进行详细描述和分析，包括数据的类型、属性、关系等。在英雄联盟数据库中，数据需求分析如下：

* + 1. 比赛数据

比赛数据包括比赛的时间、持续时间、参与的英雄、比赛结果等信息。

比赛数据的类型包括日期时间、持续时间、字符串等。

比赛数据与玩家数据和团队数据存在关联，因为比赛涉及到多个玩家和团队。

比赛数据的完整性约束包括比赛时间不能为未来时间、持续时间不能为负值等。

* + 1. 玩家数据

玩家数据包括玩家的ID、姓名、游戏表现数据（击杀数、死亡数、助攻数、金币数、小兵击杀数等）、拥有的英雄和皮肤等信息。

玩家数据的类型包括整数、字符串、布尔值等。

玩家数据与比赛数据和团队数据存在关联，因为玩家参与比赛并组成团队。

玩家数据的完整性约束包括玩家ID不能重复、游戏表现数据不能为负值等。

* + 1. 游戏元素数据

游戏元素数据包括英雄、物品和符文的信息，包括名称、属性、效果等。

游戏元素数据的类型包括字符串、数值、文本等。

游戏元素数据与比赛数据和玩家数据存在关联，因为它们都与玩家的游戏表现和比赛结果相关。

游戏元素数据的完整性约束包括元素ID不能重复、名称不能重复等。

* + 1. 团队数据

团队数据包括团队的ID、成员ID、角色等信息。

团队数据的类型包括整数、字符串等。

团队数据与玩家数据和比赛数据存在关联，因为团队由玩家组成，并参与比赛。

团队数据的完整性约束包括团队ID不能重复、成员ID不能重复等。

* + 1. 匹配系统和排名数据

匹配系统和排名数据包括匹配ID、玩家ID、匹配历史、排名比赛结果等信息。

匹配系统和排名数据的类型包括整数、字符串、文本等。

匹配系统和排名数据与比赛数据和玩家数据存在关联，因为它们记录了玩家的比赛历史和排名结果。

匹配系统和排名数据的完整性约束包括匹配ID不能重复、玩家ID不能重复等。

* 1. 业务规则及完整性约束分析

在设计英雄联盟数据库时，需要考虑各种业务规则和完整性约束，以确保数据的准确性、完整性和一致性。下面是对业务规则和完整性约束的详细分析：

* + 1. 比赛规则

参与团队人数限制： 每个团队参与比赛的玩家数量不能超过5人，且至少有两个团队参与比赛。

比赛结果规则： 每场比赛必须有明确的结果，即胜利或失败。

* + 1. 玩家规则

唯一玩家ID： 玩家ID必须是唯一的，不允许重复。

游戏表现数据非负性： 玩家的游戏表现数据，如击杀数、死亡数、助攻数、金币数、小兵击杀数等，必须为非负整数。

皮肤唯一性： 一个皮肤只能属于一个玩家，不允许重复拥有相同的皮肤。

* + 1. 游戏元素规则

唯一元素ID： 游戏元素的ID必须是唯一的，不允许重复。

价格非负性： 物品和符文的价格必须为非负数。

* + 1. 团队规则

唯一团队ID： 团队ID必须是唯一的，不允许重复。

成员ID唯一性： 一个玩家只能属于一个团队，同一个玩家ID不能出现在多个团队中。

* + 1. 匹配系统和排名规则

唯一匹配ID： 匹配ID必须是唯一的，不允许重复。

玩家ID唯一性： 一个玩家在匹配系统中只能有一个ID，同一个玩家ID不能出现在多个匹配记录中。

* + 1. 数据完整性约束

数据完整性： 数据库中的所有数据必须满足完整性约束，即不能存在冗余数据、数据必须完整且准确。

1. 数据库概念设计
   1. 确定基本实体类和属性

根据实验报告的要求，我们需要设计一个英雄联盟数据库，包括以下基本实体类和属性：

1. 英雄 (Heroes)

英雄ID (Hero\_ID) - 主键

英雄名称 (Hero\_Name)

种族 (Race)

职业 (Profession)

定位 (Position)

技能 (Skills) - 包括英雄的技能名称、描述、冷却时间等属性  
 英雄ID和英雄名称组合成联合唯一索引

2. 物品 (Items)

物品ID (Item\_ID) - 主键

物品名称 (Item\_Name)

类型 (Type)

价格 (Price)

属性 (Attributes)

效果 (Effect) - 描述物品的效果，如增加攻击力、法术强度等

3. 符文 (Runes)

符文ID (Rune\_ID) - 主键

符文名称 (Rune\_Name)

属性 (Attributes)

效果 (Effect)

4. 比赛 (Matches)

比赛ID (Match\_ID) - 主键

比赛时间 (Match\_DateTime)

比赛持续时间 (Duration)

结果 (Result)

胜利队伍ID (Winning\_Team\_ID) - 记录胜利的队伍ID

失败队伍ID (Losing\_Team\_ID) - 记录失败的队伍ID

5. 玩家 (Players)

玩家ID (Player\_ID) - 主键

玩家姓名 (Player\_Name)

击杀数 (Kills)

死亡数 (Deaths)

助攻数 (Assists)

金币数 (Gold)

小兵击杀数 (Minion\_Kills)

英雄精通等级 (Hero\_Mastery\_Level)

当前英雄ID (Current\_Hero\_ID) - 记录当前使用的英雄ID

6. 团队 (Teams)

团队ID (Team\_ID)

团队名称 (Team\_Name)

成员ID (Player\_ID) - 外键参考玩家表

角色 (Role)

Team\_ID和Player\_ID构成联合主键  
 Player\_ID设计成唯一索引

7. 皮肤 (Skins)

皮肤ID (Skin\_ID) - 主键

皮肤名称 (Skin\_Name)

是否解锁 (Unlocked)

进度或成就 (Progress\_Achievements)

8. 匹配系统和排名 (Matchmaking\_and\_Ranking)

匹配ID (Matchmaking\_ID) - 主键

玩家ID (Player\_ID) - 外键参考玩家表

匹配历史 (Match\_History)

排名比赛结果 (Ranking\_Match\_Results)

9. 比赛详情 (Match\_Details)

比赛ID (Match\_ID) - 外键参考 Match

玩家ID (Player\_ID) - 外键参考 Players

英雄ID (Hero\_ID) - 外键参考 Heroes

击杀数 (Kills)

死亡数 (Deaths)

助攻数 (Assists)

其中Match\_ID和Player\_ID构成联合主键  
 Match\_ID和Hero\_ID设置为唯一索引

10.玩家构建物品(Player\_Item)

比赛ID (Match\_ID) - 外键参考Match

玩家ID (Player\_ID) - 外键参考 Players

英雄ID (Hero\_ID) - 外键参考 Heroes

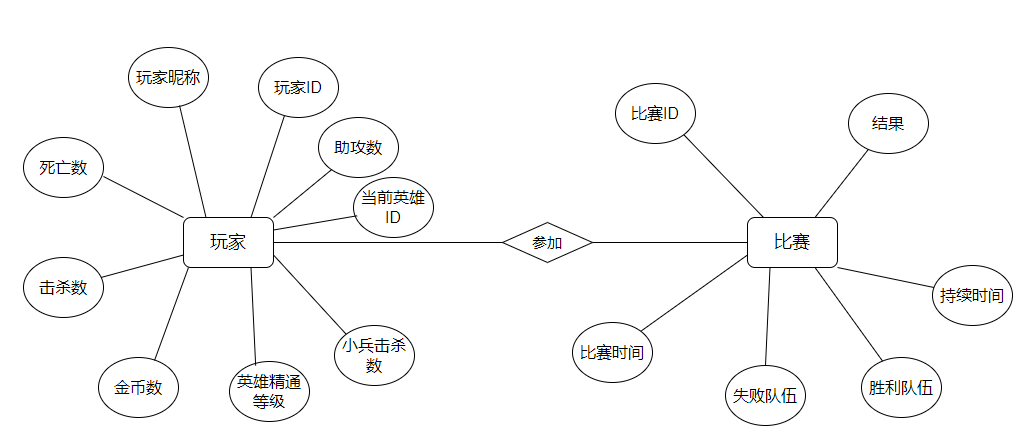
其中Match\_ID和Player\_ID和Hero\_ID构成联合主键

2.2主要业务局部概念建模

1.比赛信息记录和管理

比赛信息记录涉及**玩家**等基本实体集，而**比赛**的相关信息由玩家在对局中产生，因此需要单独建模为依赖实体集，属性有：比赛ID (Match\_ID)、比赛时间 (Match\_DateTime)、比赛持续时间 (Duration)、结果 (Result)、胜利队伍ID (Winning\_Team\_ID)、失败队伍ID (Losing\_Team\_ID)，其中比赛持续时间、结果、胜利队伍ID、失败队伍ID，应该在比赛结束之后得出，属于派生属性。

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **含 义** | **类 别** | **域 及 约 束** |
| **Match\_ID** | 比赛ID | 主键 | char(15),不允许取空值 |
| Match\_DateTime | 比赛时间 |  | datetime，不允许取空值 |
| Duration | 比赛持续时间 | 派生 | String Type，比赛结束后生成 |
| Result | 结果 | 派生 | Integer Type |
| Winning\_Team\_ID | 胜利队伍ID | 派生 | char(15),不允许取空值 |
| Losing\_Team\_ID | 失败队伍ID | 派生 | char(15),不允许取空值 |

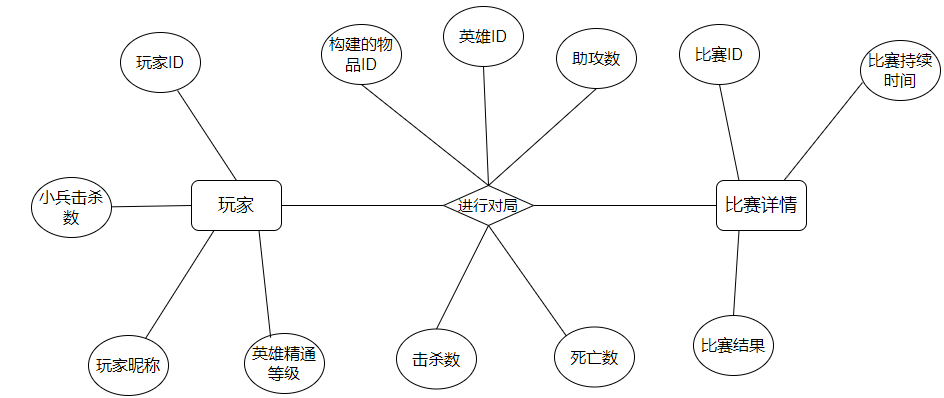


比赛信息记录和管理业务的建模

2.玩家游戏表现追踪

玩家游戏表现应该包括击杀数、死亡数、助攻数等，游戏表现都从比赛中得到，因此建模为依赖实体集，属性有玩家ID (Player\_ID) 、玩家姓名 (Player\_Name)、击杀数 (Kills)、死亡数 (Deaths)、助攻数 (Assists)、金币数 (Gold)、小兵击杀数 (Minion\_Kills)、英雄精通等级 (Hero\_Mastery\_Level)、当前英雄ID (Current\_Hero\_ID) 、记录当前使用的英雄ID、构建的物品ID (Item\_ID)，其中击杀数，死亡数，助攻数由玩家在对局中产生，属于派生属性，同样的构建的物品ID也是由玩家在对局中购买装备产生，属于派生属性

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **含 义** | **类 别** | **域 及 约 束** |
| Player\_ID | 玩家ID | 主键 | char(15),不允许取空值 |
| Player\_Name | 玩家姓名 |  | varchar(15),不允许取空值 |
| Kills | 击杀数 | 派生 | Integer Type，由对局中产生 |
| Deaths | 死亡数 | 派生 | Integer Type，由对局中产生 |
| Assists | 助攻数 | 派生 | Integer Type，由对局中产生 |
| Hero\_Mastery\_Level | 英雄精通等级 |  | char(7),取值范围:{'A','B','C','D','E','F',G'} |
| Current\_Hero\_ID | 当前英雄ID | 派生 | integer Type，由玩家选择 |
| Item\_ID | 构建的物品ID | 派生 | varchar(15),由玩家在对局中购买 |



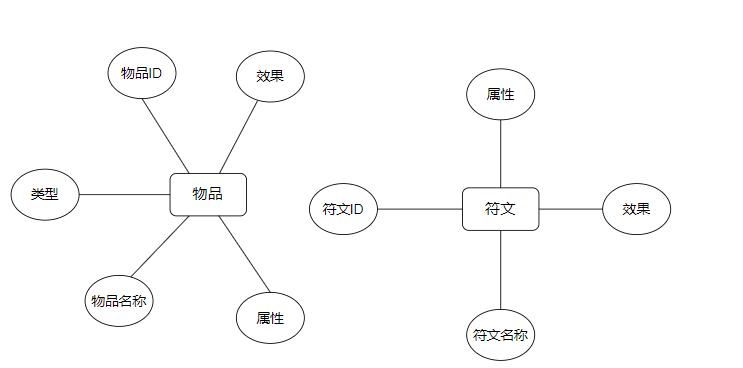
玩家游戏表现追踪业务建模

其中玩家实体集与比赛详情实体集存在多对多的进行对局联系集，联系属性有：构建的物品ID、英雄ID、助攻数、击杀数、死亡数，这些都属于派生属性，根据玩家在对局中的实际表现和操作产生。

3.物品与符文目录维护

物品与符文目录维护是指维护游戏中所有可用物品和符文的目录，包括它们的属性和效果。物品属性有：物品ID (Item\_ID)、物品名称 (Item\_Name)、类型 (Type)、价格 (Price)、属性 (Attributes)、效果 (Effect)，符文属性则有：符文ID (Rune\_ID)、符文名称、(Rune\_Name)、属性 (Attributes)、效果 (Effect)。

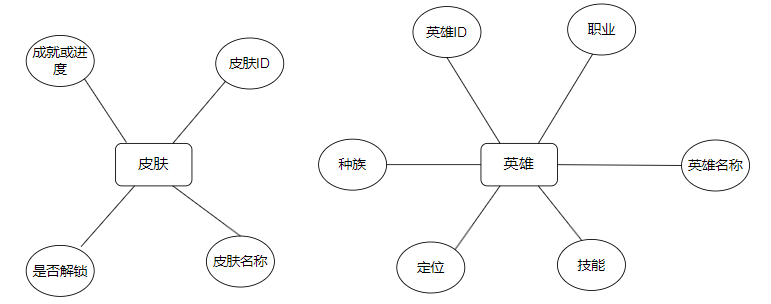
|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **含 义** | **类 别** | **域 及 约 束** |
| Item\_ID | 物品ID | 主键 | char(15),不允许取空值 |
| Item\_Name | 物品名称 |  | varchar(15),不允许取空值 |
| Type | 类型 |  | String Type，由物品类型决定 |
| Price | 价格 |  | Integer Type，不允许取空值 |
| Attributes | 属性 |  | Integer Type，不允许取空值 |
| Effect | 效果 |  | Integer Type，不允许取空值 |
| Rune\_ID | 符文ID | 主码 | char(15),不允许取空值 |
| Rune\_Name | 符文名称 |  | varchar(15),不允许取空值 |
| Attributes | 属性 |  | Integer Type，不允许取空值 |
| Effect | 效果 |  | Integer Type，不允许取空值 |



4. 玩家英雄、皮肤以及相关进度和成就追踪

该业务需要跟踪玩家拥有的英雄、皮肤以及相关的进度和成就。其中包括的属性有：英雄ID (Hero\_ID)、英雄名称 (Hero\_Name)、种族 (Race)、职业 (Profession)、定位 (Position)、技能 (Skills) - 包括英雄的技能名称、描述、冷却时间等属性，与皮肤相关的属性有：皮肤ID (Skin\_ID) - 主键、皮肤名称 (Skin\_Name)、是否解锁 (Unlocked)、进度或成就 (Progress\_Achievements)

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名** | **含 义** | **类 别** | **域 及 约 束** |
| Hero\_ID | 英雄ID | 主键 | char(15),不允许取空值 |
| Hero\_Name | 英雄名称 |  | varchar(15),不允许取空值 |
| Race | 种族 |  | varchar，不允许取空值 |
| Profession | 职业 |  | varchar，不允许取空值 |
| Position | 定位 |  | varchar，不允许取空值 |
| Skills | 技能 |  | varchar，不允许取空值 |
| Skin\_ID | 皮肤ID | 主键 | char(15),不允许取空值 |
| Skin\_Name | 皮肤名称 |  | varchar(15),不允许取空值 |
| Unlocked | 是否解锁 |  | Boolean Type，选择是或者否 |
| Progress\_Achievements | 进度或成就 |  | char(7),取值范围:{'A','B','C','D','E','F',G'} |



玩家英雄、皮肤以及相关进度和成就追踪业务局部建模

2.3定义联系集及属性

基于前面设计得到的实体集，可确定如下联系集：

（1）参加比赛（Match）联系集：他是玩家实体集和比赛实体集之间的一对多的联系集，其描述属性有比赛时间 (Match\_DateTime)、比赛持续时间 (Duration)、结果 (Result)、胜利队伍ID (Winning\_Team\_ID)、失败队伍ID (Losing\_Team\_ID)。

（2）进行对局联系集：他是玩家实体集和比赛详情实体集之间的一对多的联系集，其描述属性有=击杀数 (Kills)、死亡数 (Deaths)、助攻数 (Assists)、金币数 (Gold)、小兵击杀数 (Minion\_Kills)、=当前英雄ID (Current\_Hero\_ID)。

（3）装备联系集：他是英雄实体集和符文实体集之间一对一的联系集，没有描述属性。

（4）穿戴联系集：他是英雄实体集和皮肤实体集之间一对一的联系集，没有描述属性。

（5）购买联系集：他是玩家实体集和物品实体集之间一对一的联系集，没有描述属性。

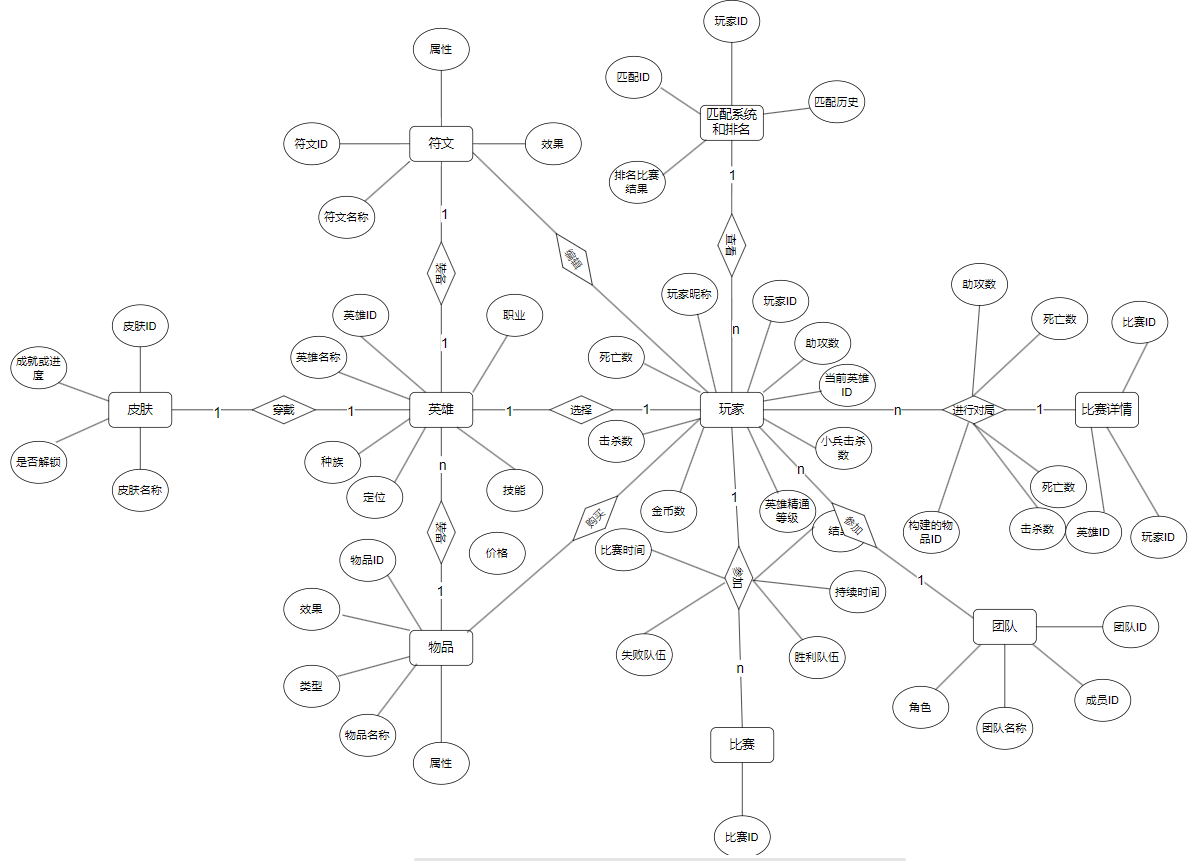
（6）查看联系集：他是玩家实体集和比赛详情实体集之间一对一的联系集，没有描述属性。

（7）参加联系集：他是玩家实体集和团队实体集之间多对一的联系集，没有描述属性。

（8）选择联系集：他是玩家实体集和英雄实体集之间一对一的联系集，没有描述属性。

2.4 完整E-R模型

综上所述，包括全部实体集、联系集及其描述属性的E-R图如下图所示。



英雄联盟数据库总E-R图

2.4 检查是否满足需求

经检查，上图所示总E-R图已经基本包含了全部需求信息描述。以下是对这些业务定义的检查：

1.比赛信息记录和管理：

联系集包括比赛和参与玩家，可以记录和管理比赛的相关信息。

属性包括比赛类型和参与玩家，满足了记录和管理比赛信息的需求。

2.玩家游戏表现追踪：

联系集包括玩家和游戏表现，可以追踪玩家在游戏中的表现。

属性包括游戏表现，可以记录玩家的胜率、击杀数等数据，满足了追踪玩家游戏表现的需求。

3.物品与符文目录维护：

联系集包括物品和符文，可以维护游戏中可用物品和符文的目录。

属性包括属性和效果，可以描述物品和符文的特征和影响，满足了维护物品和符文目录的需求。

4.玩家英雄、皮肤以及相关进度和成就追踪：

联系集包括玩家、英雄、皮肤、进度和成就，可以追踪玩家在游戏中的英雄、皮肤、进度和成就。

属性包括英雄、皮肤、进度和成就，满足了追踪玩家游戏进度和成就的需求。上述的业务定义和相关属性可以满足基本的数据记录和管理需求。

3.数据库逻辑设计

（1）英雄Heroes表：由**英雄**（Heroes）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Hero\_ID** | char(15) | 英雄ID |
| Hero\_Name | varchar(15) | 英雄名称 |
| Profession | Varchar(50) | 职业 |
| Race | Varchar(50) | 种族 |
| Position | Varchar(50) | 定位 |
| Skills | Varchar(25) | 技能 |

（2）物品Items表：由**物品**（Items）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Item\_ID** | char(15) | 物品ID |
| Item\_Name | varchar(15) | 物品名称 |
| Price | Int | 价格 |
| Attributes | Int | 属性 |
| Type | Varchar(50) | 类型 |
| Effect | Int | 效果 |

（3）符文Runes表：由**符文**（Runes）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Rune\_ID** | char(15) | 符文ID |
| Rune\_Name | varchar(15) | 符文名称 |
| Attributes | Int | 属性 |
| Effect | Int | 效果 |

（4）皮肤Skins表：由**皮肤**（Skins）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Skin\_ID** | char(15) | 皮肤ID |
| Skin\_Name | varchar(15) | 皮肤名称 |
| Unlocked | Bool | 是否解锁 |
| Progress\_Achievements | char(7) | 成就或进度 |

（5）玩家Players表：由**玩家**（Players）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Player\_ID** | char(15) | 玩家ID |
| Player\_Name | varchar(15) | 玩家昵称 |
| Current\_Hero\_ID | char(15) | 当前英雄ID |
| Hero\_Mastery\_Level | char(7) | 英雄精通等级 |
| Kills | Int | 击杀数 |
| Deaths | Int | 死亡数 |
| Assists | Int | 助攻数 |
| Minion\_Kills | Int | 小兵击杀数 |
| Gold | Int | 金币数 |

（6）比赛Matches表：由**比赛**（Matches）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Match\_ID** | char(15) | 比赛ID |
| Match\_DateTime | datetime | 比赛时间 |
| Losing\_Team\_ID | char(15) | 失败队伍ID |
| Winning\_Team\_ID | char(15) | 胜利队伍ID |
| Duration | Int | 持续时间 |
| Result | Int | 结果 |

（7）匹配Matchmaking系统：由**匹配**（Matchmaking）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Matchmaking\_ID** | char(15) | 匹配ID |
| *Player\_ID* | char(15) | 玩家ID |
| Match\_History | Int[10] | 匹配历史 |
| Ranking\_Match\_Results | Int | 排名比赛结果 |

（8）比赛详情Match\_Details表：由**比赛详情**（Match\_Details）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Match\_ID** | char(15) | 比赛ID |
| ***Player\_ID*** | char(15) | 玩家ID |
| *Hero\_ID* | char(15) | 英雄ID |
| Kills | Int | 击杀数 |
| Deaths | Int | 死亡数 |
| Assists | Int | 助攻数 |

（9）团队Teams表：由**团队**（Teams）实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Team\_ID** | char(15) | 团队ID |
| ***Player\_ID*** | char(15) | 成员ID |
| Team\_Name | varchar(15) | 团队名称 |
| Role | varchar | 角色 |

1. 玩家构建物品Player\_Item表：有玩家物品实体集转化而来

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 属性描述 |
| **Match\_ID** | char(15) | 比赛ID |
| ***Player\_ID*** | char(15) | 玩家ID |
| **Item\_ID** | char(15) | 物品ID |