Valorant analyzer

Team fightFight

> Pramod Rao Budramane | 2020111012
> Vidit Jain | 2020101134
> A Kishore Kumar | 2020101126

Introduction

Inspired by **Counter Strike Global Offensive**, Riot Games released another hit game called **Valorant**, a fast-paced first-person shooter, where it not only tests your aims and strategy but also your ability usage. Like any other e-sport game, there's a plethora of statistics generated from every action, and a variety of users are very eager to get their hands on the data.

Purpose

To store information about each round in each match in a database that can be queried by various users for their specific use case to observe the correlation between the available data points.

Users

- **Casual Users**
- **E-sport Analysts**
- **Valorant game developers**

Applications

- Can be used by casual users to analyze their own performance, whether it be in one match, their agent-specific statistics, or lifetime statistics as well.
- Can be used by professional tournaments to figure out the current meta and develop strategies.
- The people responsible for the balancing of the game can use the statistics and analysis derivable from this database system to figure out which agents need a buff and which abilities need to be nerfed in the next patch.

Database requirements

Entities

Player

A player refers to the account registered by the user via Riot. It contains user info and aggregate information over all matches they've played

Primary key: (Name, Tag) A tuple of two key attributes act as a composite key for this table

Super key : (Name, Tag, Date of Birth) A tuple of three attributes act as a super key, as it uniquely identifies a player but has redundant information, the date of birth.

Attributes

- **Name**

  Required, VARCHAR, Composite Key

- **Tag** (Every Riot account requires that the user also have a 4 digit tag. The combination of name and tag is the unique Riot ID)

  Required, VARCHAR, Composite Key

- **Date of birth**

  Required, DATE

- **Region** (Every riot account is linked to some region. Regions are used to restrict matchmaking)

  Required, VARCHAR

  Domain: [BR, EMEA, JP, KR, LATAM, NA, SEA]

- **Unlocked agents** (A multivalued attribute describing the agents playable by the player)

  Required, VARCHAR

  Domain: [Brimstone, Viper, Omen, Killjoy, Cypher, Sova, Sage, Phoenix, Jett, Reyna, Raze, Breach, Skye, Yoru, Astra, KAY/O]

- **Total time played** (in seconds)

  Required, INT

- **Rank Rating** (Each player has a RR based on which his rank is derived)

  Required, INT

- **Rank + Tier** (Composite derived attribute)

  Derived, VARCHAR

Teams (Weak Entity)

A team is a weak entity that lists the collection of all the players who are queueing together for a particular match

Composite Key: (Player 1, Player 2, Player 3, Player 4, Player 5) A tuple of the 5 player IDs

Attributes

- **Players [1, 2, 3, 4, 5]** (Composite attribute contains each player's ID)

  Required, VARCHAR, Foreign Key
- **Wins** (Total wins across all matches queued by this team)

  Required, INT
- **Number of matches played** (Total number of matches played by this team)

  Required, INT
- **Win rate** (Percentage wins per match played for this team)

  Derived, FLOAT
- **Average Rating** (Derived from the RR of each individual player)

  Required, INT
- **Coach** (Can be null, a player can choose to be the coach of a team)

  INT, Foreign Key

Matches

A match describes the interaction between players competing against each other in teams. The team playing as the attacking side wins if they kill all defenders / explode the spike. The defender side wins if they stop the plant / defuse the spike / kill all attackers.

Primary Key: Match ID (Every match is given a unique match ID)

Attributes

- **Match ID**

  Required, INT, Primary Key
- **Match start time**

  Required, INT
- **Duration** (The duration of the match in seconds)

  Required, INT
- **Team 1 + Team 2** (Composite attribute listing the team IDs of the 2 teams participating)

  Required, INT, Foreign Key
- **Score 1 + Score 2** (Composite attribute listing the scores of team 1 and team 2 respectively)

  Required, INT
- **MVP** (Most Valuable Player ID)

  Required, INT, Foreign Key
- **Winner** (Derived attribute from scores & teams, lists winning team ID)

  Required, INT, Foreign Key

- **Loser** (Derived attribute from scores & teams, lists losing team ID)

  Required, INT, Foreign Key
- **Type** (Match can either be unranked or rated)

  Required, VARCHAR, Subclass

  Domain: [RANKED, UNRANKED]
- **Map**

  Required, VARCHAR

  Domain: [ascent, icebox, bind, breeze, split, fracture, haven]

Subclass

Rated

- **RR delta** (The amount of change in RR gained/lost due to the match)

  Required, INT

Rounds (Weak Entity)

The round contains a dump of all the per round stats logged per match for a given player.

Composite Key: (Match ID, Player ID, Round Number) a tuple of 3 items guaranteed to be unique

Attributes:

- **Match ID** (The ID of the match this round belongs to)

  Required, INT, Foreign Key
- **Player ID** (The ID of the player this round shows stats of)

  Required, VARCHAR, Foreign Key
- **Round Number**

  Required, INT
- **Kills**

  Required, INT
- **Deaths**

  Required, INT
- **Assists**

  Required, INT
- **Total damage dealt**

  Required, INT
- **First blood**

  Required, BOOL
- **Planter**

  Required, BOOL
- **Defuser**

  Required, BOOL

Agent

An agent is a character that the player chooses to play in a match. Each agent has unique abilities and ultimates. Valorant has 15 agents which come under 4 classes.

Primary key: Agent ID (Each agent has a unique ID)

Attributes

- **Agent ID**

  Required, INT, Candidate Key, Primary Key

- **Name**

  Required, VARCHAR, **Candidate Key, Alternate Key**

  Domain: [Brimstone, Viper, Omen, Killjoy, Cypher, Sova, Sage, Phoenix, Jett, Reyna, Raze, Breach, Skye, Yoru, Astra, KAY/O]

- **Class**

  Required, VARCHAR

  Domain: [Controller, Sentinel, Duelist, Initiator]

- **Lore** (the backstory of the agent)

  Required, VARCHAR

Abilities (Weak entity)

Attributes:

- **Agent ID** (The ID of the agent the ability belongs to)

  Required, VARCHAR, Foreign Key

- **Name**

  Required, VARCHAR

- **Description**

  Required, VARCHAR

- **Base damage**

  Required, INT

- **Ability duration** (in seconds)

  Required, INT

- **Type** (Abilities can either be a signature ability or an ultimate ability)

  Required, VARCHAR, Subclass

  Domain: [SIGNATURE, ULTIMATE]

Subclass:

Signature

- **Cost**

  Required, INT

- **Maximum buy limit**

Required, INT

- **Cooldown** (in seconds)

  Required, INT

Ultimate

- **Ultimate points cost**

  Required, INT

- **Voice lines**

  Multi-value list, VARCHAR, Required

Relationships

Player's agents

Relationship between a **PLAYER** and a playable **AGENT** representing the matches played by a player with a certain agent. It provides the wins, the number of matches played, win rate, total kills, total deaths, total assists, KDA, total damage dealt, total credits spent, average ECON rating.

Cardinality ratio: (P : A) (Player : Agent)

Participation constraints: Partial participation

It is not required for every player to be associated with at least one agent (although it is very likely). Hence, partial participation from players.

It is not required for every agent to be played by at least one player (although it is very likely). Hence, partial participation from agents.

Relationship type: Binary relationship between two strong entities

Attributes:

- Wins
- Number of matches played
- Win-rate (Derived)
- Total kills
- Total deaths
- Total assists
- KDA (Derived) (Total kills and assists per death)
- Total damage dealt
- Total credits spent
- Average ECON rating (Derived) (Total damage dealt per 1000 credits spent)

Agent's abilities

Relationship between an **AGENT** and the agent's **ABILITIES**. Every agent has a set of abilities (includes signature and ultimate abilities).

Cardinality ratio: (1 , 1) : (4 , 4) (Agent : Abilities)

Participation constraints: Total participation

Each agent has 3 signature abilities and 1 ultimate ability.

Relationship type: Binary relationship between a strong and a weak entity.

Match stats

Relation between a **PLAYER** who has played in a certain **MATCH** and the performance of that player in the **ROUNDS** that ensued.

Cardinality constraints: (10, 10) : (1, M) : (13, 25) (Player : Match : Round)

Participation constraints:

Partial participation from players.

T**otal participation from Match and Rounds.**

It is not required for every player to have played at least one match (although it is very likely). Hence, partial participation from players.

For every match and round, there need to be 10 players. Hence, total participation from the match and Round.

Relationship type: Ternary relationship between (strong - strong - weak) entities.

Modelling 3-degree relationship as binary relationships:

- **PLAYER** plays in a **ROUND**
- **MATCH** consists of **ROUNDS**
- **PLAYER** participates in a **MATCH**

Coaching

A player who has signed up as the coach for some team can coach the players in that team. A player can sign up as coach for multiple teams. This is a relationship from a **PLAYER** to a **PLAYER** where the two participating players have distinct roles.

Cardinality constraints: 1 : 1 (PLAYER as a coach : PLAYER as a participating player)

Participation constraints: Partial participation.

Every player need not be a coach for some player. Hence, partial participation from player as a coach.

Every player does not need to be coached. Hence, partial participation from the player as a participating player.

Relationship type: Unary relationship between two strong entities of the same type, but with *distinct roles*.

Five-stack stats

Relationship between a **PLAYER** and a **TEAM** that the player has played with. It provides the wins, the number of matches played, win rate, total kills, total deaths, total assists, KDA, total damage dealt, total credits spent, average ECON rating.

Cardinality ratio: (5, 5) : (1, T) (Player : Team)

Participation constraints: Partial participation from PLAYER and complete participation from TEAM.

It is not required for every player to be associated with at least one team (in case the player has not played any matches). Hence, partial participation from players.

Every team must be associated with exactly 5 players. Hence, complete participation from the team.

Relationship type: Binary relationship between a strong and weak entity (Strong - Weak)

Attributes:

- Total kills
- Total deaths
- Total assists
- KDA (Derived) (Total kills and assists per death)
- Total damage dealt
- Total credits spent
- Average ECON rating (Derived) (Total damage dealt per 1000 credits spent)

Match description

Relationship describing the details of a match. It relates a **MATCH** played between two participating **TEAMS,** the **PLAYERS** in those teams, the **AGENTS** played by those players.

Cardinality ratio: (1, 1) : (2, 2) : (10, 10) : (5, 10) (Match : Team : Player : Agent)

In each match, the players of opposing teams might choose the same agents making the agent cardinality ratio for the agents (5, 10).

Participation constraints:

Total participation from **MATCH a**nd **TEAMS.**

Partial participation from **PLAYERS** and **AGENTS.**

Relationship type: It is a 4-ary relationship between (strong - weak - strong - strong) entities.

Modelling 4-degree relationship as binary relationships:

- **MATCH** is played between two **TEAMS**
- **TEAM** contains **PLAYERS**
- **PLAYERS** play an **AGENT**
- **AGENTS** are picked in a **MATCH**

Functional Requirements

The database model allows us to not only organize our data but also perform queries on the data in a simple and orchestrated manner that will allow us to easily analyze the data stored. We achieve this by implementing functional query capabilities which allow us to easily retrieve and analyze data as required.

Retrieval

Selection

- Get the list of all teams the player has been a part of
- Get the list of all the matches a team has played
- Get the list of all the round stats of a player for a particular match
- Get the list of all the matches played between two different teams

- Get a list of all the agents and their abilities

Projection

- Get a list of all players with K/D ≥ x
- Get a list of all agents for a specific player with K/D ≥ x
- Get a list of all agents for a specific player with a win rate ≥ x

Aggregate

- Total wins & losses of a player
- Total wins & losses of a player for a particular agent
- Average win rate of a player in a particular map

Search

- Partial term search match for agent & ability
- Partial term search match for player name

Analysis

- **Player report -** Total wins, matches played, win-rate, total damage over all time, win-rate on each map, win-rate with each agent & econ rating as managed by the player.
- **Team report -** Total wins, matches played, win-rate, win-rate on each map.
- **Agent report -** Total wins over all matches played, win rate, total damage dealt and econ-rating as maintained by all players who use a particular agent