

# **LoL Tournament**

Alberto Cherobin, Davide Spaziante

## **ABSTRACT**

League Of Legends (che d'ora in avanti chiameremo LoL per semplicità) è un videogioco online, pubblicato il 27 ottobre 2009, sviluppato da Riot Games, del genere MOBA. I MOBA (Multiplayer Online Battle Arena) sono un sottogenere dei videogiochi strategici in tempo reale. In una mappa chiusa, due squadre, arroccate nelle proprie basi (che contano diversi edifici e il quartier generale - "QG"), hanno l'obiettivo di distruggere il QG avversario e preservare il proprio. Ciascun giocatore, una volta creato l'account, partirà dal livello 1; tale livello aumenterà grazie all'esperienza ricavata dalle partite giocate e non esiste un numero massimo di livello raggiungibile. Le partite possono essere di due tipologie: normale, dove ciascun giocatore può crescere di esperienza, capendo a fondo le meccaniche del gioco e classificata, alla quale si ha accesso solo una volta raggiunto il livello 30, che permette al giocatore di avere un punto di riferimento per capire il proprio livello di abilità raggiunto. In ogni partita ci sono due squadre composte da 5 giocatori ciascuna, ognuno dei quali ricopre un ruolo ben preciso. L'evocatore, una volta trovata una partita, si trova a dover scegliere quale campione utilizzare, dipendentemente dal ruolo che deve ricoprire, scegliendo da una pool in continuo aumento e che ad oggi conta oltre 100 personaggi acquistabili. Una volta scelto, il giocatore ha la possibilità (se lo possiede) di selezionare un aspetto alternativo da quello base per il proprio campione, due incantesimi sui 9 disponibili e la runa chiave che andrà ad influire sulle sue statistiche durante la partita. Tutti i giocatori possono premiare il compagno che ritengono più meritevole a fine partita incrementando il suo livello di onore. I personaggi e gli aspetti sono tutte acquistabili sia con essenze, ovvero punti ricavati giocando, oppure con soldi veri e propri. A distanza di 8 anni LoL conta milioni e milioni di giocatori, diventando uno dei giochi più famosi nel suo genere. Vengono organizzati tornei di tutte le categorie e vanta una coppa del mondo annuale in cui si scontrano le migliori squadre di ogni continente. "LoL Tournament" è un base di dati che raccoglie le informazioni relative agli account dei membri di una società che crea squadre per tornei a vari livelli di competizione, cercando anche di descrivere tutti gli aspetti che rappresentano questo gioco, ovvero rune, incantesimi e mosse di ciascun personaggio. I gestori della società andranno ad analizzare le informazioni relative a ciascun giocatore, quali la lega di appartenenza, i campioni più giocati e le prestazioni recenti, in modo da andare a comporre nel miglior modo possibile una squadra.

## **DESCRIZIONE DEI REQUISITI**

Si vuole realizzare una base di dati che contenga le informazioni relative a giocatori che desiderano partecipare a tornei di League Of Legends, in modo da comporre squadre con giocatori con stesso livello di abilità e ognuno specializzato in un determinato ruolo nella partita. Vogliamo anche essere in grado di analizzare la capacità di un giocatore nel costruire un proprio equipaggiamento in gioco e le scelte relative a rune e incantesimi per sfruttare al meglio le capacità del personaggio che andrà ad utilizzare. Ciascun utente possiede un proprio account di gioco, identificato univocamente dal nome, di cui vogliamo sapere la data di creazione, il suo livello attuale e l'onore, che rispecchia il comportamento del giocatore verso i compagni di squadra, che parte dal livello base 2 fino a 5. Vogliamo anche sapere se appartiene ad un club. Ogni campione è caratterizzato da un nome, che lo identifica univocamente. Su di esso vogliamo sapere la data in cui è stato rilasciato, il costo in essenze, ovvero la moneta ottenibile giocando e anche il costo in denaro che, come per le essenze, è definito esclusivamente dalla sua data di uscita. Un aspetto delicato per gli sviluppatori del gioco è quello di attribuire delle statistiche base equilibrate a ciascun personaggio; tali statistiche sono il movimento, l'attacco fisico, il potere magico e la difesa. Ogni campione può essere inserito in un genere specifico, ovvero: tank, mago, assassino, tiratore oppure supporto. Nel campo di battaglia sono presenti 4 diverse zone di combattimento che ciascuna squadra dovrà gestire al meglio: la corsia superiore, la corsia centrale e la corsia inferiore. Esse sono collegate dal quarto luogo della mappa, ovvero la giungla. Ciascun campione ha una posizione principale in cui riesce ad esprimere al meglio le proprie capacità, e, anche se non tutti, una corsia "secondaria" in cui può essere utilizzato. Ognuno di loro possiede 4 mosse e per ciascuna di esse ci interessa il nome, che la identifica univocamente, il tasto che permette al giocatore di attivare quella mossa (Q, W, E o R) e la descrizione della mossa, ovvero cosa accade una volta premuto il tasto. Ciascuna mossa infligge danni fisici o magici, oppure entrambi, perciò vogliamo conoscere i danni base di ciascuna mossa. Di ogni campione sono presenti più versioni sotto forma di aspetti, di ognuno di essi vogliamo sapere il nome, che lo identifica univocamente, il campione a cui appartiene e la categoria a cui appartiene (classica, epica, leggendaria e suprema), che ne determina il costo. Le classiche possono costare 5 oppure 8 euro, le epiche 10 oppure 12 euro, le leggendarie 14 oppure 16 euro ed infine le supreme che costano 20 euro. È importante anche la data di rilascio dell'aspetto. Per ciascun campione

posseduto da un giocatore vogliamo sapere: il nome del campione, il nome dell'utente che lo possiede ed anche la maestria su di esso. Quest'ultima indica quanto il giocatore abbia utilizzato il personaggio. Le maestrie in totale sono di 4 tipi: maestria di livello 4 (significa che il campione è stato utilizzato in 30 partite), maestria di livello 5 (significa che il campione è stato utilizzato in 50 partite), maestria di livello 6 (significa che il campione è stato utilizzato in 70 partite) e maestria di livello 7 (significa che il campione è stato utilizzato in 100 partite). Vogliamo sapere anche la data di acquisizione del campione e in che modalità è stato acquistato, se con essenze o con soldi veri. Per ciascun aspetto posseduto vogliamo sapere il proprietario dell'aspetto, il suo nome che lo identifica, la data in cui è avvenuto l'acquisto e la modalità con cui è stato acquistato, se con essenze o con soldi veri. Per ciascun giocatore vogliamo tenere conto delle ultime 4 partite che ha disputato, in particolare vogliamo sapere per ciascuna: chi ha disputato la partita, la data e l'ora in cui è avvenuta, l'esito (vittoria o sconfitta), il nome del campione e del possibile aspetto utilizzato. (Assumiamo che i giocatori non si incontrino mai nelle 4 partite registrate quindi, anche se due o più partite hanno la stessa data e ora, non sono la stessa partita). È importante anche sapere il voto finale che riflette la prestazione del giocatore in partita e le possibilità sono: S (eccellente), A (ottima), B (sufficiente), C (insufficiente). Nel caso in cui la partita risulti essere di tipologia classificata ci interessa sapere quanti punti ha guadagnato il giocatore in caso di vittoria oppure quanti ne ha persi in caso di sconfitta. Durante la partita il giocatore accumula dell'oro che può essere speso per comprare oggetti per andare a potenziare le statistiche del proprio campione. Durante la selezione del campione l'evocatore può scegliere, oltre alla runa chiave, due incantesimi obbligatori tra i 10 disponibili. Di ciascun oggetto vogliamo sapere il suo nome che lo identifica univocamente, il costo in gioco e la tipologia a cui appartiene, ovvero se difesa, attacco, magia oppure movimento. Ovviamente ciascun oggetto a seconda della tipologia andrà ad incrementare quella particolare statistica. Per esempio un oggetto di tipologia "attacco" andrà ad incrementare l'attacco fisico del campione. Di ciascun incantesimo vogliamo sapere il nome che lo identifica univocamente e una breve descrizione che spiega in cosa consiste. Una volta selezionata la runa chiave, essa incrementa le statistiche iniziali di quel campione. Dunque per ciascuna Runa, oltre al nome, vogliamo sapere quanto aggiunge in Movimento, in Attacco Fisico, in Potere Magico e in Difesa. All'interno del gioco è possibile creare dei club. Non ha nessuna utilità ai fini del gioco, semplicemente permette ai giocatori, facenti parte dello stesso club, di identificarsi in quella che potrebbe essere una squadra con il proprio nome e il proprio tag. Per ogni club dunque vogliamo sapere: il nome che lo identifica univocamente, la data in cui è stato creato, il giocatore capo del club e il tag visibile una volta in gioco. In League Of Legends le partite classificate si suddividono in due tipi: classificate solo/duo, in cui un giocatore potrà partecipare ad una partita da solo oppure al massimo insieme ad un amico, e le classificate flex, in cui si può giocare da soli oppure insieme ad un massimo di quattro amici. In base all'esito delle 10 partite di piazzamento, (non trattate nella base di dati), in uno dei due tipi di classificate il giocatore viene inserito in una lega, dalla 1 (la più alta) alla 4 (la più bassa), ciascuna costituita da 5 divisioni. Una volta piazzato per la prima volta in uno dei due tipi di classifiche si viene inseriti in un gruppo assieme ad altri giocatori senza distinzione di lega e si rimane nello stesso gruppo anche se si effettuano i piazzamenti nell'altra classifica, ad esempio: il giocatore "pippo" effettua le partite di piazzamento in solo/duo, e viene classificato Lega 2 divisione 2 nel gruppo dei "tornadi di akali"; un mese dopo decide di effettuare le partite di piazzamento anche in flex, e viene classificato Lega 3 divisione 5, sempre nel gruppo dei "tornadi di akali". Interessa dunque sapere, per ogni giocatore presente nel database, nel caso abbia effettuato uno dei due piazzamenti, il nome della classifica, e sia per la classifica Flex, sia per quella Solo/Duo, vogliamo sapere il numero della lega, il numero della divisione, dalla 1, ovvero la più alta, alla 5, ovvero la più bassa. Per passare da una divisione ad un'altra bisogna arrivare a 100 punti, i quali vengono attribuiti dopo aver vinto una partita classificata, mentre in caso di sconfitta ci sarà una conseguente perdita di punti.

## **ANALISI DEI REQUISITI**

### **Frase relative a "LoL Tournament"**

Si vuole realizzare una base di dati che contenga le informazioni relative a giocatori che desiderano partecipare a tornei di League Of Legends, in modo da comporre squadre con giocatori dello stesso livello di abilità e ognuno specializzato in un determinato ruolo nella partita.

#### **Frase relative a "Giocatore/Utente"**

Ciascun utente possiede un proprio account di gioco, identificato univocamente dal nome, di cui vogliamo sapere la data di creazione, il suo livello attuale e l'onore, che rispecchia il comportamento del giocatore verso i compagni di squadra, che parte dal livello base 2 fino a 5. Vogliamo anche sapere se appartiene ad un club.

#### **Frase relative a "Statistiche Campione"**

Un aspetto delicato per gli sviluppatori del gioco è quello di attribuire delle statistiche base equilibrate a ciascun personaggio; tali statistiche sono il movimento, l'attacco fisico, il potere magico e la difesa.

#### **Frase relative a "Campione/Personaggio"**

Ogni campione è caratterizzato da un nome, che lo identifica univocamente. Su di esso vogliamo sapere la data in cui è stato rilasciato, il costo in essenze, ovvero la moneta ottenibile giocando e anche il costo in denaro che, come per le essenze, è definito esclusivamente dalla sua data di uscita.

#### **Frase relative a "Informazioni generali del campione"**

Ogni campione può essere inserito in un genere specifico, ovvero: tank, mago, assassino, tiratore oppure supporto. Nel campo di battaglia sono presenti 4 diverse zone di combattimento che ciascuna squadra dovrà gestire al meglio: la corsia superiore, la corsia centrale e la corsia inferiore. Esse sono collegate dal quarto luogo della mappa, ovvero la giungla. Ciascun campione ha una posizione principale in cui riesce ad esprimere al meglio le proprie capacità, e, anche se non tutti, una corsia "secondaria" in cui può essere utilizzato.

#### **Frase relative a "Mossa campione"**

. Ognuno di loro possiede 4 mosse e per ciascuna di esse ci interessa il nome, che la identifica univocamente, il tasto che permette al giocatore di attivare quella mossa (Q, W, E o R) e la descrizione della mossa, ovvero cosa accade una volta premuto il tasto. Ciascuna mossa infligge danni fisici o magici, oppure entrambi, perciò vogliamo conoscere i danni base di ciascuna mossa.

#### **Frase relative ad "Aspetto"**

Di ogni campione sono presenti più versioni sotto forma di aspetti, di ognuno di essi vogliamo sapere il nome, che lo identifica univocamente, il campione a cui appartiene, la categoria a cui appartiene (classica, epica, leggendaria e suprema), il costo, che in questo caso dipende dalla tipologia dell'aspetto [...]

#### **Frase relative a "Campione Posseduto"**

Per ciascun campione posseduto da un giocatore, e per posseduto si intende che lo abbia acquistato in modo permanente dal negozio, vogliamo sapere: il nome del campione, il nome dell'utente che lo possiede ed anche la maestria su di esso. Quest'ultima indica quanto il giocatore abbia utilizzato il personaggio. [...]

#### **Frase relative ad "Aspetto posseduto"**

Per ciascun aspetto posseduto da ciascun giocatore vogliamo sapere il proprietario dell'aspetto, il nome che identifica l'aspetto, la data in cui è avvenuto l'acquisto e la modalità con cui l'aspetto è stato acquistato, con essenze oppure con soldi veri.

#### **Frase relative a "Partita"**

Per ciascun giocatore vogliamo tenere conto delle ultime 4 partite che ha disputato, in particolare vogliamo sapere per ciascuna: il nome utente di colui che ha disputato la partita, la data e l'ora in cui è avvenuta, l'esito (vittoria o sconfitta) e ovviamente il nome del campione che quel determinato utente ha utilizzato in quella partita. È importante anche sapere il voto finale che riflette la prestazione del giocatore in partita [...]

#### **Frase relative a "Classificata"**

Nel caso in cui la partita risulti essere di tipologia classificata ci interessa sapere quanti punti ha guadagnato il giocatore in caso di vittoria oppure quanti ne ha persi in caso di sconfitta.

#### **Frase relative a "Equipaggiamento"**

Durante la partita il giocatore accumula dell'oro che può essere speso per comprare oggetti per andare a potenziare le statistiche del proprio campione. Durante la selezione del campione l'evocatore può scegliere, oltre alla runa chiave, due incantesimi obbligatori tra i 10 disponibili.

### Frasi relative ad "Oggetto"

Di ciascun oggetto disponibile in gioco vogliamo sapere il suo nome che lo identifica univocamente, il suo costo in gioco, la tipologia a cui appartiene, ovvero difesa, attacco, magia oppure movimento. Ovviamente ciascun oggetto a seconda della tipologia andrà ad incrementare una particolare statistica. Per esempio un oggetto di tipologia "attacco" andrà ad incrementare l'attacco fisico del campione che ne entra in possesso.

### Frasi relative ad "Incantesimo"

Di ciascun incantesimo vogliamo sapere il nome che lo identifica univocamente e una breve descrizione che spiega in cosa consiste.

### Frasi relative a "Runa"

Una volta selezionata la runa chiave, essa incrementa le statistiche iniziali di quel campione. Dunque per ciascuna Runa, oltre al nome, vogliamo sapere quanto aggiunge in Movimento, in Attacco, in Magia e in Difesa.

### Frasi relative a "Club"

All'interno del gioco è possibile creare dei club. Non ha nessuna utilità ai fini del gioco, semplicemente permette ai giocatori, facenti parte dello stesso club, di identificarsi in quella che potrebbe essere una squadra con il proprio nome e il proprio tag. Per ogni club dunque vogliamo sapere: il nome che lo identifica univocamente, la data in cui è stato creato, il giocatore capo del club e il tag visibile una volta in gioco.

### Frasi relative a "Classifiche"

Interessa dunque sapere, per ogni giocatore presente nel database, nel caso abbia effettuato uno dei due piazzamenti, il nome della classifica, e sia per la classifica Flex, sia per quella Solo/Duo, vogliamo sapere il numero della lega, il numero della divisione, dall'1, ovvero la più alta, al 5, ovvero la più bassa. Per passare da una divisione ad un'altra bisogna arrivare a 100 punti, i quali vengono attribuiti dopo aver vinto una partita classificata, mentre in caso di sconfitta ci sarà una conseguente perdita di punti.

## PROGETTAZIONE CONCETTUALE

### ▪ Entità e Attributi

#### • Utente/Evocatore/Giocatore

- Nome: varchar
- Livello: int
- Data iscrizione: date
- Livello onore: int
- Club: varchar

#### • Campione

- Nome: varchar
- Data del Rilascio: date
- Costo (attributo composto):
  - In essenze: int
  - In denaro: int

#### • Mossa campione

- Nome: varchar
- Tasto: char
- Danni Magici: int
- Danni Fisici: int
- Descrizione: varchar
- Campione: varchar

#### • Informazioni campione

- Campione: varchar
- Genere: varchar
- Corsia principale: varchar

- Corsia secondaria: varchar

#### • Aspetto

- Nome: varchar
- Data uscita: date
- Costo (attributo composto):
  - In essenze: int
  - In denaro: int
- Campione possessore: varchar
- Tipologia: varchar

#### • Campione posseduto

- Possessore: varchar
- Nome campione: varchar
- Maestria: int
- Data acquisizione: date
- Modalità acquisto: varchar

#### • Aspetto Posseduto

- Proprietario: varchar
- Nome: varchar
- Data Acquisizione: date
- Modalità acquisto: varchar

#### • Partita

- Evocatore: varchar
- Cronologia: datetime
- Esito: varchar

- Campione usato: varchar
- Voto: char
- Aspetto usato: varchar
- **Equipaggiamento**
  - Giocatore: varchar
  - Cronologia: datetime
  - Oggetto uno: varchar
  - Oggetto due: varchar
  - Oggetto tre: varchar
  - Oggetto quattro: varchar
  - Runa chiave: varchar
  - Incantesimo uno: varchar
  - Incantesimo due: varchar
- **Incantesimo**
  - Nome: varchar
  - Descrizione: varchar
- **Runa**
  - Nome: varchar
  - Incremento Movimento: int
  - Incremento Difesa: int
  - Incremento Attacco Fisico: int
  - Incremento Potere Magico: int
- **Oggetto**
  - Nome: varchar
  - Costo: int

- Tipologia: varchar
- Statistiche aggiuntive: int
- **Classifiche**
  - Utente: varchar
  - Nome classifica: varchar

#### **Generalizzazione totale di Classifica:**

- **Solo/Duo**
  - Numero divisione: int
  - Numero punti: int
  - ID\_Lega: int
- **Flex**
  - Numero divisione: int
  - Numero punti: int
  - ID\_Lega: int

#### **Generalizzazione parziale su Partita:**

- **Classificata**
  - Cronologia: datetime
  - Evocatore: varchar
  - Punti: int
  - Tipologia: varchar

### ▪ **Relazioni**

- Giocatore - Campione posseduto: acquisto campione
  - Ogni giocatore può possedere da 0 a N campioni.
  - Ogni campione posseduto appartiene ad 1 solo giocatore.
- Campione - Aspetto: versione
  - Ogni campione può avere da 1 versione, ovvero quella base, fino ad N versioni.
  - Ogni aspetto è una versione dell'aspetto base di uno e un solo campione.
- Giocatore - Aspetto posseduto: acquisto aspetto
  - Ogni giocatore può possedere da 0 ad N aspetti.
  - Ogni aspetto posseduto appartiene ad 1 solo giocatore.
- Campione - Campione posseduto: istanza di campione
  - Ogni campione posseduto è istanza di 1 solo campione.
  - Ogni campione può avere da 0 ad N istanze di se stesso.
- Aspetto - Aspetto posseduto: istanza di aspetto
  - Ogni aspetto posseduto è istanza di 1 solo aspetto.
  - Ogni aspetto può avere da 0 ad N istanze di se stesso.
- Giocatore - Partite: presenza
  - Ogni giocatore può effettuare da 1 fino ad N partite.
  - In una partita sono presenti 10 giocatori.
- Giocatore - Classifiche: appartenenza
  - Ogni giocatore appartiene ad 1 solo gruppo di classifica.
  - Ad ogni gruppo di classifica possono appartenere da 0 ad N giocatori.
- Partita – Equipaggiamento: potenziamento
  - Ogni giocatore all'interno di una partita può crearsi 1 solo equipaggiamento da 4 oggetti.
  - All'interno di una partita ci sono 10 equipaggiamenti per ciascuno dei 10 giocatori.

- Equipaggiamento – Runa: specifiche runa
  - In ogni equipaggiamento è presente 1 sola runa.
  - In ogni partita può esserci 1 sola runa, nel caso tutti i 10 giocatori scegliessero la stessa, fino ad un massimo di tutte e 9, nel caso 9 giocatori su 10 scegliessero 9 rune diverse.
- Equipaggiamento – Incantesimo: specifiche incantesimo
  - In ogni equipaggiamento ci sono sempre 2 incantesimi.
  - In una partita, nel caso in cui tutti i giocatori scegliessero gli stessi incantesimi, avremmo solo questi 2 ma potrebbero comparire anche tutti e 10 se almeno 5 giocatori scegliessero una coppia di incantesimi l'una diversa dall'altra.
- Equipaggiamento – Oggetto: specifiche oggetto
  - Ogni equipaggiamento può avere da 0 fino ad un massimo di 4 oggetti.
  - In ogni partita possono essere presenti da 0 fino ad un massimo di N oggetti.
- Campione – Mossa campione: capacità
  - Ogni campione ha 4 mosse a disposizione.
  - Ogni mossa appartiene ad 1 solo campione.
- Campione – Informazione campione: dettagli
  - Ogni campione ha un insieme di caratteristiche non modificabili dal rilascio.
  - Determinate informazioni possono appartenere da 1 fino ad N campioni. In quanto esistono personaggi con informazioni identiche ma che differiscono per altre meccaniche di gioco.
- Club – Giocatore: iscrizione
  - Ogni giocatore può appartenere a 0 o ad 1 solo club.
  - Ogni club può contenere da 1 fino a N giocatori.
- Campione – Statistiche campione: potenziale
  - Ogni campione può avere solo 1 sola serie di statistiche.
  - Una serie di statistiche possono appartenere ad 1 campione fino ad un massimo di N.

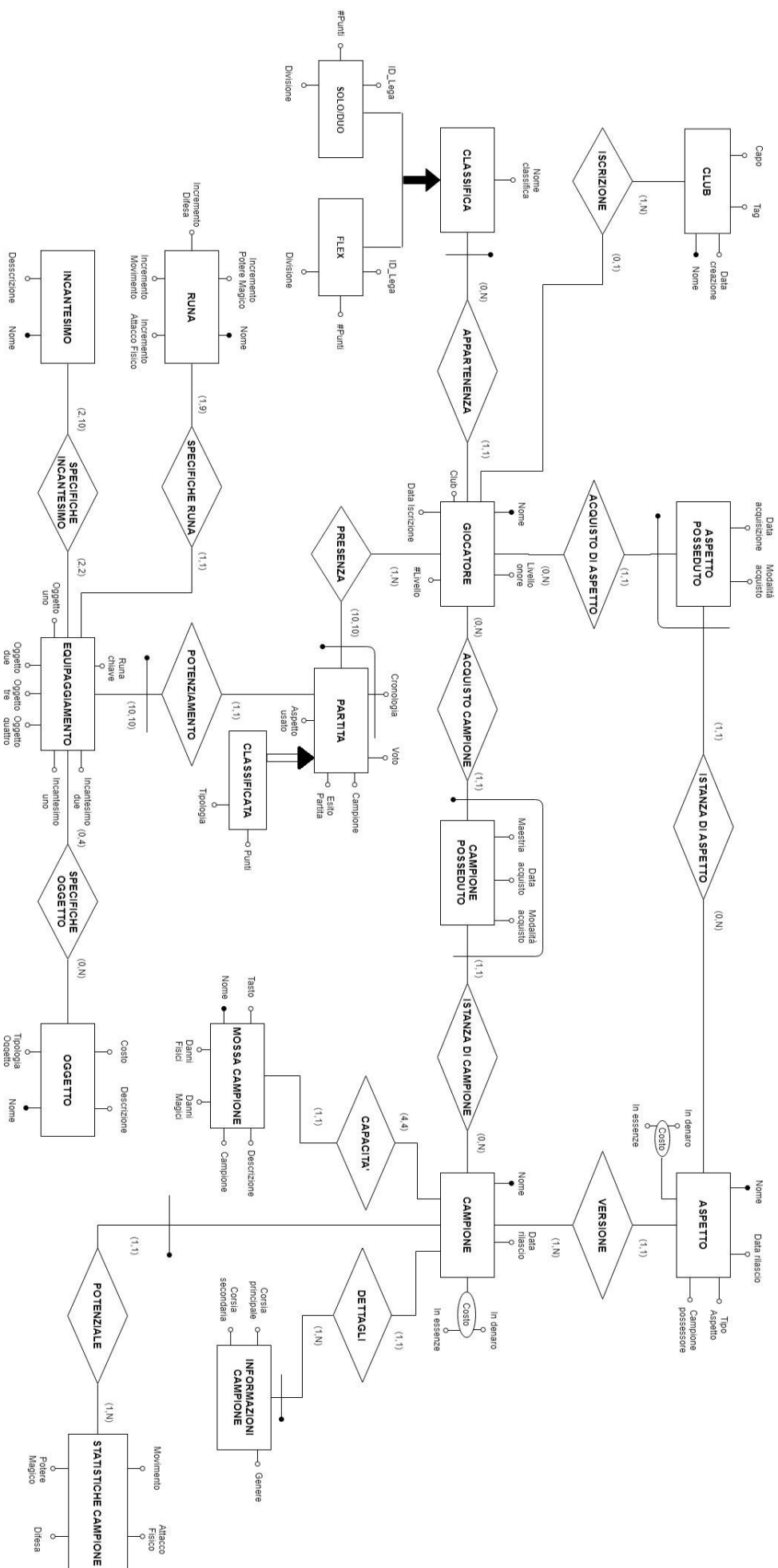
## **SCHEMA ER E SCHEMA LOGICO**

Nelle prossime due facciate sono presenti lo schema ER e a seguire lo schema ER ristrutturato. In quest'ultimo sono stati apportati i seguenti cambiamenti:

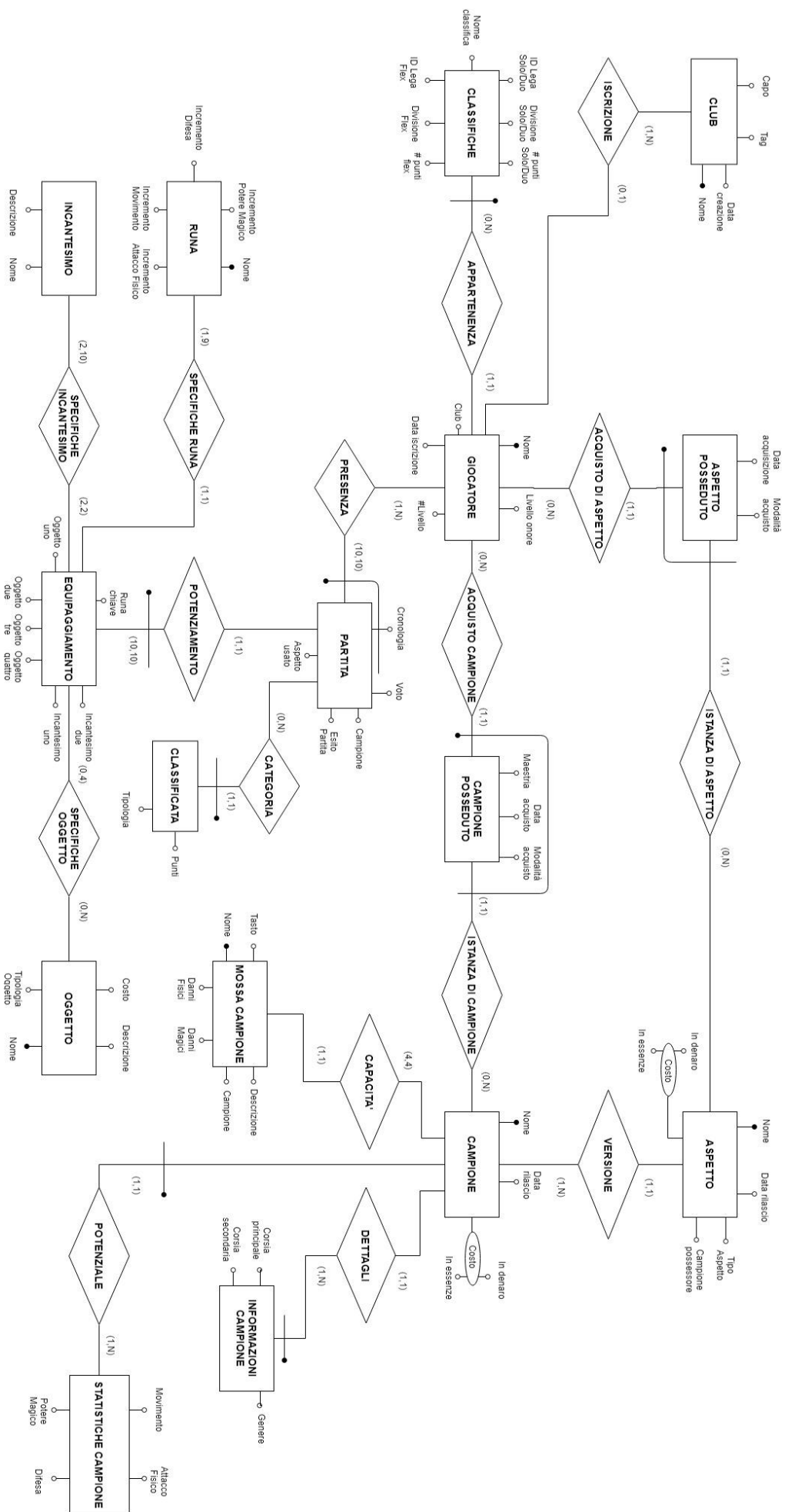
1) È stata eliminata la generalizzazione totale sull'entità "Classifica". Le entità figlie "Solo/Duo" e "Flex" sono state accorpate nell'entità genitore e loro proprietà sono state aggiunte a quest'ultima. I nomi delle proprietà sono stati leggermente modificati in modo da poter distinguere a quale delle due entità figlie prima appartenessero.

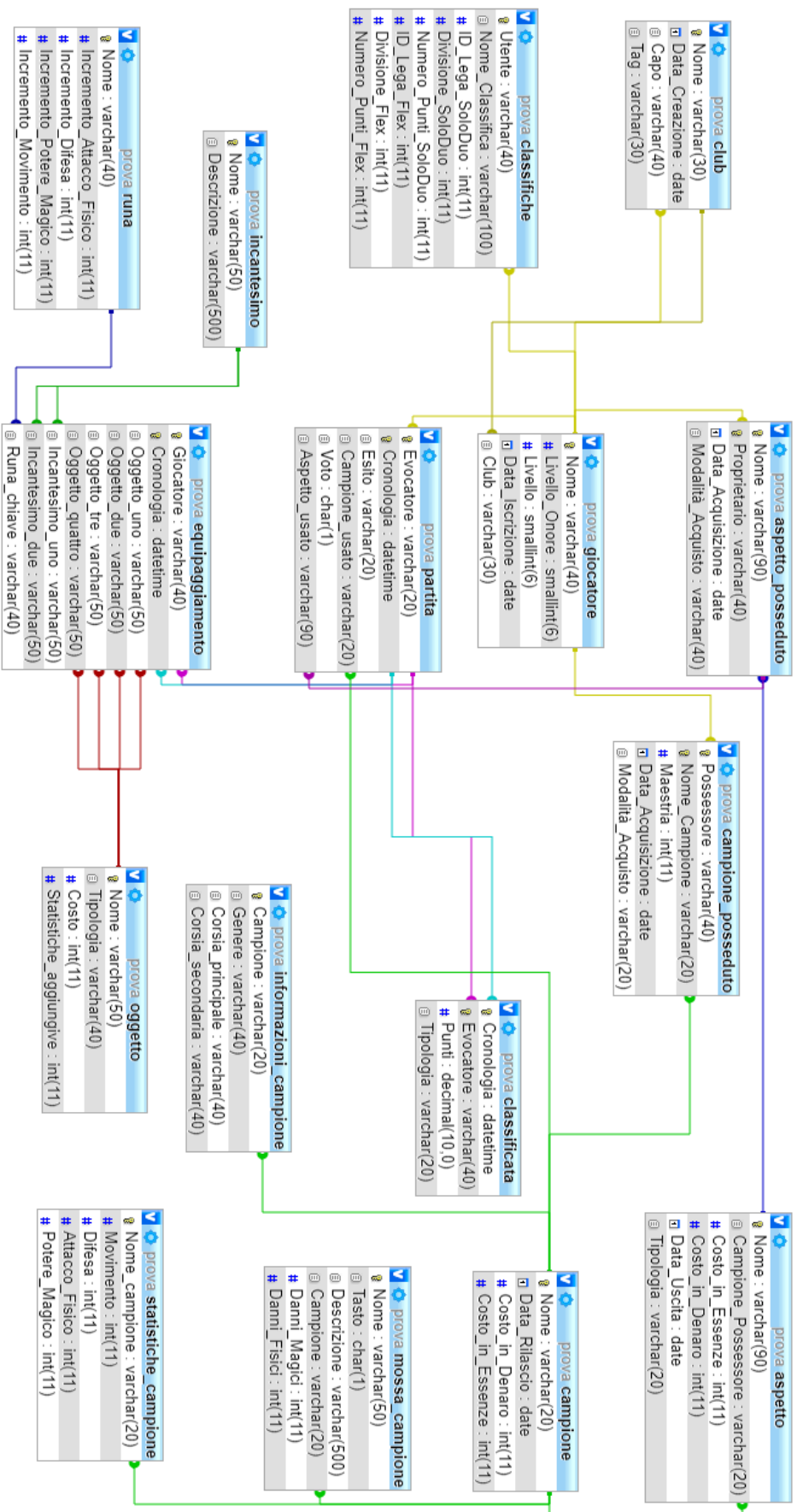
2) È stata eliminata la generalizzazione parziale sull'entità "Partita". Una nuova relazione ora collega quest'ultima con l'entità "Classificata". In questo modo si può chiaramente distinguere l'andamento di un giocatore nelle partite "normali" rispetto a quelle di tipologia "classificata".

Dopo i due schemi ER è stato inserito lo schema logico che va ad evidenziare i vincoli intrarelazionali ma soprattutto i vincoli interrelazionali, quali le chiavi esterne.









## SCHEMA RELAZIONALE

**GIOCATORE** (Nome, Livello onore, Livello, Data iscrizione, Club)

**CLUB** (Nome, Data creazione, Capo, Tag)

**CAMPIONE** (Nome, Data rilascio, Costo in denaro, Costo in essenze)

**MOSSA CAMPIONE** (Nome, Tasto, Descrizione, Campione, Danni Magici, Danni Fisici)

**INFORMAZIONI CAMPIONE** (Nome campione, Genere, Corsia principale, Corsia secondaria)

**ASPETTO** (Nome, Campione possessore, Costo in denaro, Costo in essenze, Data uscita, Tipologia)

**ASPETTO POSSEDUTO** (Proprietario, Nome, Data acquisizione, Modalità acquisto)

**CAMPIONE POSSEDUTO** (Possessore, Nome, Maestria, Data acquisto, Modalità acquisto)

**PARTITA** (Evocatore, Cronologia, Esito, Campione usato, Voto, Aspetto usato)

**CLASSIFICATA** (Evocatore, Cronologia, Punti, Tipologia)

**CLASSIFICHE** (Utente, ID Lega Solo/Duo, Divisione Solo/Duo, #Punti Solo/Duo, ID Lega Flex, Divisione Flex, #Punti Flex)

**EQUIPAGGIAMENTO** (Giocatore, Cronologia, Oggetto uno, Oggetto due, Oggetto tre, Oggetto quattro, Runa chiave, Incantesimo uno, Incantesimo due)

**OGGETTO** (Nome, Costo, Tipologia, Statistiche aggiuntive)

**INCANTESIMO** (Nome, Descrizione)

**RUNA** (Nome, Incremento attacco fisico, Incremento potere magico, Incremento difesa, Incremento movimento)

**STATISTICHE CAMPIONE** (Nome campione, Movimento, Difesa, Attacco fisico, Potere magico)

**FUNZIONE 1:** Vogliamo creare una funzione che, dati in input data e ora di una partita di un giocatore (cioè l'attributo "Cronologia" dell'entità "Partita" o "Equipaggiamento") e una statistica a scelta tra "Attacco", "Magia", "Movimento" o "Difesa", ci calcoli il valore di quest'ultima in base al campione usato, all'equipaggiamento creato e alla runa scelta.

**CREATE FUNCTION** Calcolo\_Statistica\_A\_Scelta(PartitaGiocata DATETIME, Statistica **VARCHAR(60)**)

**RETURNS INT**

**BEGIN**

**DECLARE** AttaccoFisicoIncrementato **INT**;

**DECLARE** PotereMagicoIncremento **INT**;

**DECLARE** DifesaIncrementata **INT**;

**DECLARE** MovimentoIncrementato **INT**;

**DECLARE** Oggetti **INT**;

**IF**(Statistica = "Attacco") **THEN**

**SET** AttaccoFisicoIncrementato = (**SELECT** s.Attacco\_Fisico

**FROM** equipaggiamento **as** e **JOIN** partita **as** p **ON** p.Cronologia = e.Cronologia

**JOIN** statistiche\_campione **as** s **ON** s.Nome\_campione = p.Campione\_usato

**WHERE** e.Cronologia = PartitaGiocata) +

(**SELECT** r.Incremento\_Attacco\_Fisico

**FROM** equipaggiamento **as** e **JOIN** partita **as** p **ON** p.Cronologia = e.Cronologia

**JOIN** runa **as** r **ON** e.Runa\_chiave = r.Nome

**WHERE** e.Cronologia = PartitaGiocata);

**SET** Oggetti = (**SELECT SUM**(o.Statistiche\_aggiungive)

**FROM** equipaggiamento **as** e **JOIN** oggetto **as** o **ON** o.Nome = e.Oggetto\_Uno **OR** o.Nome = e.Oggetto\_due **OR** o.Nome = e.Oggetto\_tre **OR** o.Nome = e.Oggetto\_quattro

**WHERE** o.Tipologia = "Attacco" **AND** e.cronologia = PartitaGiocata);

**IF** Oggetti **IS NULL** **THEN RETURN** AttaccoFisicoIncrementato;

**ELSE RETURN** AttaccoFisicoIncrementato + Oggetti; **END IF**;

**END IF**;

IF(Statistica = "Magia") THEN

```
SET PotereMagicoIncremento = (SELECT s.Potere_Magico
                                FROM equipaggiamento as e JOIN partita as p ON p.Cronologia = e.Cronologia
                                JOIN statistiche_campione as s ON s.Nome_campione = p.Campione_usato
                                WHERE e.Cronologia = PartitaGiocata) +
                                (SELECT r.Incremento_Potere_Magico
                                 FROM equipaggiamento as e JOIN partita as p ON p.Cronologia = e.Cronologia
                                 JOIN runa as r ON e.Runa_chiave = r.Nome
                                 WHERE e.Cronologia = PartitaGiocata);

SET Oggetti = (SELECT SUM(o.Statistiche_aggiungive)
                FROM equipaggiamento as e JOIN oggetto as o ON o.Nome = e.Oggetto_Uno OR o.Nome =
                e.Oggetto_due OR o.Nome = e.Oggetto_tre OR o.Nome = e.Oggetto_quattro
                WHERE o.Tipologia = "Magia" AND e.cronologia = PartitaGiocata);
```

IF Oggetti IS NULL THEN RETURN PotereMagicoIncremento;

ELSE RETURN PotereMagicoIncremento + Oggetti; END IF;

END IF;

IF (Statistica = "Difesa") THEN

```
SET DifesaIncrementata = (SELECT s.Difesa
                            FROM equipaggiamento as e JOIN partita as p ON p.Cronologia = e.Cronologia
                            JOIN statistiche_campione as s ON s.Nome_campione = p.Campione_usato
                            WHERE e.Cronologia = PartitaGiocata) +
                            (SELECT r.Incremento_Difesa
                             FROM equipaggiamento as e JOIN partita as p ON p.Cronologia = e.Cronologia JOIN
                             runa as r ON e.Runa_chiave = r.Nome
                             WHERE e.Cronologia = PartitaGiocata);

SET Oggetti = (SELECT SUM(o.Statistiche_aggiungive)
                FROM equipaggiamento as e JOIN oggetto as o ON o.Nome = e.Oggetto_Uno OR o.Nome =
                e.Oggetto_due OR o.Nome = e.Oggetto_tre OR o.Nome = e.Oggetto_quattro
                WHERE o.Tipologia = "Difesa" AND e.cronologia = PartitaGiocata);
```

IF Oggetti IS NULL THEN RETURN DifesaIncrementata;

ELSE RETURN DifesaIncrementata + Oggetti; END IF;

END IF;

IF (Statistica = "Movimento") THEN

```
SET MovimentoIncrementato = (SELECT s.Movimento
                              FROM equipaggiamento as e JOIN partita as p ON p.Cronologia = e.Cronologia JOIN
                              statistiche_campione as s ON s.Nome_campione = p.Campione_usato
                              WHERE e.Cronologia = PartitaGiocata) +
                              (SELECT r.Incremento_Movimento
                               FROM equipaggiamento as e JOIN partita as p ON p.Cronologia = e.Cronologia JOIN
                               runa as r ON e.Runa_chiave = r.Nome
                               WHERE e.Cronologia = PartitaGiocata);

SET Oggetti = (SELECT SUM(o.Statistiche_aggiungive)
                FROM equipaggiamento as e JOIN oggetto as o ON o.Nome = e.Oggetto_Uno OR o.Nome =
                e.Oggetto_due OR o.Nome = e.Oggetto_tre OR o.Nome = e.Oggetto_quattro
                WHERE o.Tipologia = "Movimento" AND e.cronologia = PartitaGiocata);
```

IF Oggetti IS NULL THEN RETURN MovimentoIncrementato;

ELSE RETURN MovimentoIncrementato + Oggetti; END IF;

END IF;

END;

**FUNZIONE 2:** Vogliamo creare una funzione che, dato in input data e ora della partita e una mossa del campione, identificata con uno dei 4 tasti Q, W, E o R, ci dia la quantità di danni totali che provoca se usata. Sostanzialmente, oltre alle statistiche base, il danno della mossa è dato dalla quantità di Attacco o Magia totale che un campione accumula. Se una mossa, di base, provoca più danni fisici rispetto a quelli magici, andrà sommato l'attacco fisico totale, viceversa si dovrà aggiungere il potere magico.

```
CREATE FUNCTION CalcoloDanniMossa(PartitaGiocata DATETIME, TastoMossa VARCHAR(1))  
RETURNS INT
```

```
BEGIN
```

```
    DECLARE DanniFisici INT;  
    DECLARE DanniMagici INT;
```

```
SET DanniFisici = (SELECT m.Danni_Fisici FROM mossa_campione AS m WHERE m.Tasto =  
TastoMossa AND m.Campione = ANY (SELECT partita.Campione_usato  
FROM partita  
WHERE partita.Cronologia = PartitaGiocata));
```

```
SET DanniMagici = (SELECT m.Danni_Magici FROM mossa_campione AS m WHERE m.Tasto =  
TastoMossa AND m.Campione = ANY (SELECT partita.Campione_usato  
FROM partita  
WHERE partita.Cronologia = PartitaGiocata));
```

```
IF DanniFisici > DanniMagici THEN
```

```
RETURN DanniFisici + DanniMagici + Calcolo_StatistichA_Scelta(PartitaGiocata, "Attacco");
```

```
ELSE RETURN DanniFisici + DanniMagici + Calcolo_StatistichA_Scelta(PartitaGiocata, "Magia");
```

```
END IF;
```

```
END
```

**TRIGGER 1:** Vogliamo creare un Trigger che una volta inserito un nuovo campione all'interno della base di dati, crea una nuova riga nelle tabelle "aspetto", "informazioni campione" e "statistiche campione". Crea anche 4 righe nella tabella "mosse campione" per le 4 mosse del nuovo personaggio. I gestori della base di dati in un secondo tempo provvederanno ad inserire i dati veri e propri.

```
CREATE TRIGGER Nuovo Campione
```

```
AFTER INSERT ON campione
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO mossa_campione VALUES ("Aggiungi nome mossa Q", "Q", "Aggiungi descrizione  
mossa Q", New.Nome, 0, 0);
```

```
    INSERT INTO mossa_campione VALUES ("Aggiungi nome mossa w", "W", "Aggiungi descrizione  
mossa w", New.Nome, 0, 0);
```

```
    INSERT INTO mossa_campione VALUES ("Aggiungi nome mossa E", "E", "Aggiungi descrizione  
mossa E", New.Nome, 0, 0);
```

```
    INSERT INTO mossa_campione VALUES ("Aggiungi nome mossa R", "R", "Aggiungi descrizione  
mossa R", New.Nome, 0, 0);
```

```
    INSERT INTO informazioni_campione VALUES (New.Nome, "Aggiungi Genere", "Aggiungi corsia  
principale", "Aggiungi corsia secondaria");
```

```
    INSERT INTO statistiche_campione VALUES (New.Nome, 0, 0, 0, 0);
```

```
    INSERT INTO aspetto VALUES ("Aggiungere nome aspetto", New.Nome, 0, 0, NEW.Data_Rilascio,  
"Inserire Tipologia" );
```

```
END
```

**TRIGGER 2:** Vogliamo creare un trigger che, una volta inserita una partita classificata di tipologia Flex, in base ai punti vinti o persi a fine partita, va ad aggiornare i dati nella tabella "classifiche" relativi al giocatore che ha effettuato quella partita. (Il trigger per gestire la classifica SoloDuo sarebbe identico, cambierebbero solo i nomi da Flex a SoloDuo.)

```
BEGIN
```

```
DECLARE NOME VARCHAR(30);
```

```

DECLARE PuntiAux INT;
DECLARE DivAux INT;
DECLARE LegaAux INT;

IF NEW.Tipologia = "Flex" THEN
    SET NOME = (SELECT p.Evocatore FROM partita AS p WHERE p.Cronologia = NEW.Cronologia);
    SET DivAux = (SELECT cla.Divisione_Flex FROM classifiche as cla WHERE cla.Utente =
NEW.Evocatore);
    SET PuntiAux = (SELECT cla.Numero_Punti_Flex FROM classifiche as cla WHERE cla.Utente =
NEW.Evocatore);
    SET LegaAux = (SELECT cla.ID_Lega_Flex FROM classifiche as cla WHERE cla.Utente =
NEW.Evocatore);

    IF(PuntiAux + NEW.Punti) < 0 THEN
        IF LegaAux = 5 AND DivAux = 5 THEN
            UPDATE classifiche
            SET Numero_Punti_Flex = 0
            WHERE Utente = NOME;
        ELSE
            IF DivAux = 5 THEN
                UPDATE classifiche
                SET ID_Lega_Flex = ID_Lega_Flex + 1, Numero_Punti_Flex = 0, Divisione_Flex = 1
                WHERE Utente = NOME;
            ELSE
                UPDATE classifiche
                SET Numero_Punti_Flex = 0, Divisione_Flex = Divisione_Flex + 1
                WHERE Utente = NOME;
            END IF; END IF;
        ELSE
            IF(PuntiAux + NEW.Punti) > 100 THEN
                IF (LegaAux = 1 AND DivAux = 1) THEN
                    UPDATE classifiche
                    SET Numero_Punti_Flex = 100
                    WHERE Utente = NOME;
                ELSE
                    IF DivAux = 1 THEN
                        UPDATE classifiche
                        SET ID_Lega_Flex = ID_Lega_Flex - 1, Numero_Punti_Flex = 0, Divisione_Flex = 5
                        WHERE Utente = NOME;
                    ELSE
                        UPDATE classifiche
                        SET Numero_Punti_Flex = 0, Divisione_Flex = Divisione_Flex - 1
                        WHERE Utente = NOME;
                    END IF;
                END IF;
            ELSE
                UPDATE classifiche
                SET Numero_Punti_Flex = Numero_Punti_Flex + NEW.Punti
                WHERE Utente = NOME;
            END IF; END IF;
        END IF;
    END

```

**QUERY 1:** Vogliamo sapere i primi 3 giocatori che hanno speso di più in aspetti con denaro vero in ordine decrescente.

```
CREATE VIEW Classifica_spese_aspetti AS
SELECT a.Proprietario, SUM(b.Costo_in_denaro) as Denaro_speso
FROM aspetto_posseduto AS a JOIN aspetto as b ON a.Nome = b.Nome
WHERE a.Modalità_acquisto = "Denaro"
GROUP BY a.Proprietario
ORDER BY Denaro_speso DESC LIMIT 3
```

#### OUTPUT QUERY 1

Proprietario	Denaro_speso
dracos777	52
CanonStylus	44
Hiken96	42

**QUERY 2:** Vogliamo sapere i nomi dei primi 5 campioni più comprati la cui corsia principale risulta essere quella centrale. Devono essere ordinati in maniera decrescente rispetto al numero di volte che sono stati comprati.

```
SELECT c.Nome, COUNT(cp.Possessore) as Possessori
FROM campione_posseduto as cp JOIN campione AS c ON c.Nome = cp.Nome_Campione
WHERE c.Nome in (SELECT info.Campione
                  FROM informazioni_campione as info
                  WHERE info.Corsia_principale = "Centrale")
GROUP BY c.Nome
ORDER BY Possessori DESC LIMIT 5
```

#### OUTPUT QUERY 2:

Nome	Possessori
Aurelion Sol	5
Akali	4
Yasuo	3
Ahri	3
Ekkio	3

**QUERY 3:** Vogliamo sapere quali giocatori hanno la media del livello di maestria sui loro campioni superiore al 6, in particolare di quelli classificati in Lega 1 in modalità Solo/Duo e che allo stesso tempo sono i capi di un club creato dal 2015 in poi.

```
SELECT cp.Possessore, AVG(cp.Maestria) as Media_Maestria
FROM campione_posseduto as cp
WHERE cp.Maestria >= 4 AND cp.Possessore IN (SELECT cla.Utente
                                              FROM classifiche AS cla
                                              WHERE cla.ID_Lega_SoloDuo = 1) AND
      cp.Possessore IN (SELECT club.Capo
                        FROM club
                        WHERE club.Data_Creazione > '2015-01-01')
GROUP BY cp.Possessore
HAVING Media_Maestria > 6
```

#### OUTPUT QUERY 3:

Possessore	Media_Maestria
DarkVo0doo	6.5000



**QUERY 4:** Vogliamo calcolare la media del costo degli oggetti dell'equipaggiamento delle partite il cui esito è una vittoria e i cui giocatori hanno il livello di onore al livello massimo.

```
SELECT e.Giocatore, AVG(o.Costo) as Media_costo_oggetti
FROM (equipaggiamento as e JOIN oggetto as o ON o.Nome = e.Oggetto_Uno OR o.Nome =
e.Oggetto_due OR o.Nome = e.Oggetto_tre OR o.Nome = e.Oggetto_quattro) JOIN partita AS p ON
e.Cronologia = p.Cronologia AND e.Giocatore = p.Evocatore
WHERE e.Giocatore IN (SELECT g.Nome
                      FROM Giocatore as g
                      WHERE g.Livello_Onore > 4) AND p.Esito = "Vittoria"
GROUP BY e.Giocatore
```

**OUTPUT QUERY 4:**

Giocatore	Media_costo_oggetti
Felix889	2600.0000
Hiken96	2650.0000
LaMaggica	2525.0000
Mario889	2300.0000
MirrorUp	2585.7143
Oh My Darph	2562.5000
xTigerMaster	2587.5000

**QUERY 5:** Vogliamo sapere i nomi dei giocatori che si sono iscritti al gioco dal 2013 in poi, con il livello account maggiore di 30 e classificati in modalità Solo/Duo in Lega 3. Per questi giocatori vogliamo sapere chi ha speso meno di 22000 essenze nell'acquisto di campioni.

```
SELECT cp.Possessore, SUM(c.Costo_in_Essenze) AS Somma_essenze_spese
FROM campione_posseuto AS cp JOIN campione AS c ON cp.Nome_Campione = c.Nome
WHERE cp.Modalità_Acquisto = "Essenze" AND
      cp.Possessore IN (SELECT giocatore.Nome
                       FROM giocatore
                       WHERE giocatore.Livello > 30 AND giocatore.Data_Iscrizione > "2013-01-01"
                       AND giocatore.Nome IN (SELECT classifiche.Utente
                                              FROM classifiche
                                              WHERE classifiche.ID_Lega_SoloDuo = 3 ))
GROUP BY cp.Possessore
HAVING Somma_essenze_spese < 22000
```

**OUTPUT QUERY 5:**

Possessore	Somma_essenze_spese
CanonStylus	20700

**QUERY 6:** Vogliamo sapere quante partite sono state perse da giocatori che utilizzavano degli aspetti classici oppure epici il cui campione può essere utilizzato in una corsia secondaria.

```
SELECT COUNT(p.Aspetto_usato) AS Numero_aspetti
FROM partita AS p
WHERE p.Aspetto_usato IS NOT NULL
      AND p.Aspetto_usato IN (SELECT asp.Nome FROM aspetto AS asp
                              WHERE asp.Tipologia = "Epica" OR asp.Tipologia = "Classica")
      AND p.Campione_usato IN (SELECT info.Campione FROM informazioni_campione AS info
                              WHERE info.Corsia_secondaria IS NOT NULL) AND P.Esito = "Sconfitta"
```

**OUTPUT QUERY 6:**

Numero_aspetti
3