

Progetto Basi di Dati

Red Bull Bc One

Davide Tommasin - Matricola 1073541

| | |
|--|----|
| Progetto Basi di Dati | 1 |
| Red Bull Bc One | 1 |
| 1. Abstract | 4 |
| 2. Analisi dei Requisiti | 4 |
| 3. Progettazione Concettuale | 5 |
| 3.1 Descrizione delle Classi | 5 |
| 3.2 Descrizione delle Relazioni | 6 |
| 3.3 Descrizione dei Vincoli di Integrità aggiuntivi | 7 |
| 4. Schema Entity - Relationship (E-R) | 8 |
| 5. Ristrutturazione schema E-R | 9 |
| 5.1 Eliminazione delle Generalizzazioni | 9 |
| 6. Schema Entity - Relationship (E-R) Ristrutturato | 10 |
| 7. Progettazione Logica | 11 |
| 8 Definizione dello schema logico tramite DDL di MySQL | 13 |
| 9 Query - Trigger - Funzioni | 17 |
| 9.1 Query | 17 |
| 9.1.1 Query | 17 |
| 9.1.2 Query | 17 |
| 9.1.3 Query | 18 |
| 9.1.4 Query | 18 |
| 9.1.5 Query | 19 |
| 9.1.6 Query | 20 |
| 9.2 Trigger | 20 |
| 9.2.1 Trigger | 20 |
| 9.2.2 Trigger | 21 |
| 9.3 Funzioni | 21 |
| 9.3.1 Funzione | 21 |
| 9.3.2 Funzione | 22 |
| 10 Interfaccia WEB | 23 |
| 10.1 home.php | 23 |
| 10.2 login.php | 23 |
| 10.3 verificaLogin.php | 23 |
| 10.4 registrazione.php | 23 |
| 10.5 logout.php | 24 |
| 10.6 jQuery.js | 24 |
| 10.7 homeQuery | 24 |
| 10.8 ajaxRequest.php | 24 |
| 10.9 queryProgetto.php | 24 |

| | |
|------------------------------|----|
| 10.10 triggerProgetto.php | 24 |
| 10.11 funzioniProgetto.php | 24 |
| 10.12 homeAmministratore.php | 24 |
| 10.13 visualizza.php | 25 |
| 10.14 inserisci.php | 25 |
| 10.15 modifica.php | 25 |
| 10.16 modificaRecord.php | 25 |
| 10.17 cancella.php | 25 |

1. Abstract

Red Bull BC One è un torneo a livello mondiale di break dance, il quale si svolge ogni anno sponsorizzata dall'azienda Red Bull, iniziato nel 2004 fino ad oggi.

Si tratta di un insieme di contest "one on one" (uno vs uno) in cui ballerini di break dance (chiamati b-boy), provenienti da tutto il mondo, si scontrano in gare ad eliminazione diretta, venendo giudicati da cinque esperti, cercando di vincere il contest finale.

Per vincere il Red Bull BC One ci sono tre tappe principali, vincere un contest nazionale detto cypher (solo b-boy di quella nazione potranno partecipare al contest), i quali si svolgono in città di tutto il mondo.

I vincitori dei cypher si qualificheranno per i contest continentali detti final, sono sei in tutto il mondo (latin america, north america, eastern europa, western europa, asia pacific, middle east africa).

I sei b-boy vincitori dei final si qualificheranno per il contest mondiale il world final, dove si sfideranno tra di loro e contro altri b-boy da tutto il mondo, invitati direttamente dalla Red Bull perché si sono distinti durante l'anno precedente in altri contest minori.

Il vincitore del world final sarà il campione di quella edizione del Red Bull BC One.

2. Analisi dei Requisiti

Si vuole realizzare una base di dati e la relativa applicazione web che contenga e gestisca le informazioni relative al Red Bull BC One ed in particolare si vogliono conoscere: i vincitori del contest world final delle varie edizioni.

Ogni Red Bull BC One ha una edizione che è identificata da:

- l'anno, che la identifica univocamente.

Ad ogni edizione partecipano dei b-boys, i quali si possono dividere in tre tipi:

- i b-boys invitati direttamente dalla Red Bull, i quali vengono invitati direttamente al contest world final;
- i b-boys che devono iscriversi ad un cypher;
- i giudici i quali sono b-boy esperti, che giudicheranno i vari contest.

Ogni b-boy è identificato da:

- il nickname;
- la nazionalità, che insieme al nickname lo identificano univocamente;
- il nome;
- il cognome;
- la data di nascita.

Ogni edizione del Red Bull BC One è organizzata in contest, i contest si suddividono in:

- cypher (contest nazionali), dove si possono iscrivere solo i b-boys appartenenti alla nazione dove si svolge il cypher. I cypher sono presenti in tutto il mondo e anche più di uno per nazione;
- final (contest continentali), ai quali parteciperanno solo i vincitori dei vari cypher, i contest final sono sei in tutto il mondo e si suddividono per continente, e sono:

Latin America;

North America;

Eastern Europe;

Western Europe;

Asia Pacific;

Middle East Africa.

- world final (contest mondiale), al quale parteciperanno i sei b-boys vincitori dei contest final e alcuni b-boys invitati direttamente dalla Red Bull.

Per ogni tipologia di contest (cypher, final, world final) ci sarà solo un vincitore.

Ogni contest è identificato da:

- la nazione;
- la città dove si svolge, che insieme alla nazione la identificano univocamente;
- la data di svolgimento.

Ad ogni contest suonerà un dj, il quale verrà identificato da:

- il nickname;
- la nazionalità, che insieme al nickname lo identificano univocamente;
- il nome;
- il cognome;
- la data di nascita.

3. Progettazione Concettuale

3.1 Descrizione delle Classi

-**Edizione Red Bull**: tiene traccia dell'edizione

- Anno: *year*

-**B-Boy**: contiene le informazioni relative ad un ballerino

- Nickname: *string*
- Nazionalità: *string*
- Nome: *string*
- Cognome: *string*
- Data di Nascita: *date*

Sono definite le seguenti sottoclassi di B-Boy

-**B-Boy Invitati**: i ballerini che vengono invitati direttamente al contest world final

-**B-Boy Iscritti**: i ballerini che devono iscriversi ad un contest cypher

-**Giudici**: ballerini esperti che giudicheranno i contest

-**Contest**: contiene le informazioni relative alle gare

- Nazione: *string*
- Città: *string*
- DataSvolgimento: *date*

Sono definite le seguenti sottoclassi di Contest

-**Cypher**: la prima tappa, il contest dove i b-boy si possono iscrivere

-**Final**: la seconda tappa, dove parteciperanno solo i b-boy vincitori dei cypher

Sono definite le seguenti sottoclassi di Final

-**Latin America**: contest final nel sud America

-**North America**: contest final nel nord America

-**Eastern Europa**: contest final nel est Europa

-**Western Europa**: contest final nel ovest Europa

-**Asia Pacific**: contest final in Asia e Oceania

-**Middle East Africa**: contest final nel est Africa

-**World Final**: terza e ultima tappa, dove parteciperanno i b-boy vincitori dei 6 final e i b-boy invitati direttamente dalla Red Bull

-**Dj**: contiene le informazioni del Dj che suonerà al contest

- Nickname: *string*
- Nazionalità: *string*
- Nome: *string*
- Cognome: *string*
- Data di Nascita: *date*

Oltre a queste classi ci sarà anche la classe Login che servirà per gestire gli utenti nell'interfaccia web

-**Login**: Contiene le informazioni di un utente

- Username: *string*
- Password: *string*
- email: *string*

3.2 Descrizione delle Relazioni

EdizioneRedBull-Contest: **Organizzazione**

- Ogni EdizioneRedBull è *organizzato* in 1 o più contest (1,N)
- Ogni contest *appartiene* ad una ed una sola EdizioneRedBull (1,1)

EdizioneRedBull-BBoy: **Partecipazione**

- Ad ogni EdizioneRedBull *partecipano* 1 o più b-boy (1,N)
- Ogni b-boy può *participa* 1 e 1 sola volta a quell'EdizioniRedBull (1,1)

Dj-Contest: **Suonare**

- Un Dj può *suonare* ad uno o più contest (1,N)
- Ad un contest *suona* uno e un solo Dj (1,1)

Giudice-Contest: **Presiedere**

- Un giudice può *presiedere* ad uno o più contest (0,N)
- Ad un contest *presiedono* più giudici (1,N)

BBoylscritto-Cypher: **Iscrizione**

- Un bboy si può *iscrivere* ad uno o più cypher (1,N)
- Ad un cypher si possono *iscrivere* uno o più bboy (1,N)

BBoyInvitato-WorldFinal: **Invito**

- Un bboy si può *invitato* ad un solo WorldFinal (1,1)
- Ad un WorldFinal possono essere *invitati* uno o più bboy (1,N)

Cypher-Final: **AvanzamentoCypher**

- Il vincitore di un determinato cypher *avanza* ad uno ed uno solo final (1,1)
- Ad un determinato final *avanzano* i vincitori di più cypher (1,N)

Final-WorldFinal: **AvanzamentoFinal**

- Il vincitore di un determinato final *avanza* ad uno ed uno solo world final (1,1)
- Al world final *avanzano* i vincitori di più final (1,N)

BBoyscritto-Cypher: **VittoriaCypher**

- Solo un bboy iscritto *vince* un solo cypher (0,1)
- Ad un specifico cypher *vince* uno ed un solo bboy iscritto (1,1)

BBoyscritto-Final: **VittoriaFinal**

- Solo un bboy iscritto *vince* un solo final (0,1)
- Ad un specifico final *vince* uno ed un solo bboy iscritto (1,1)

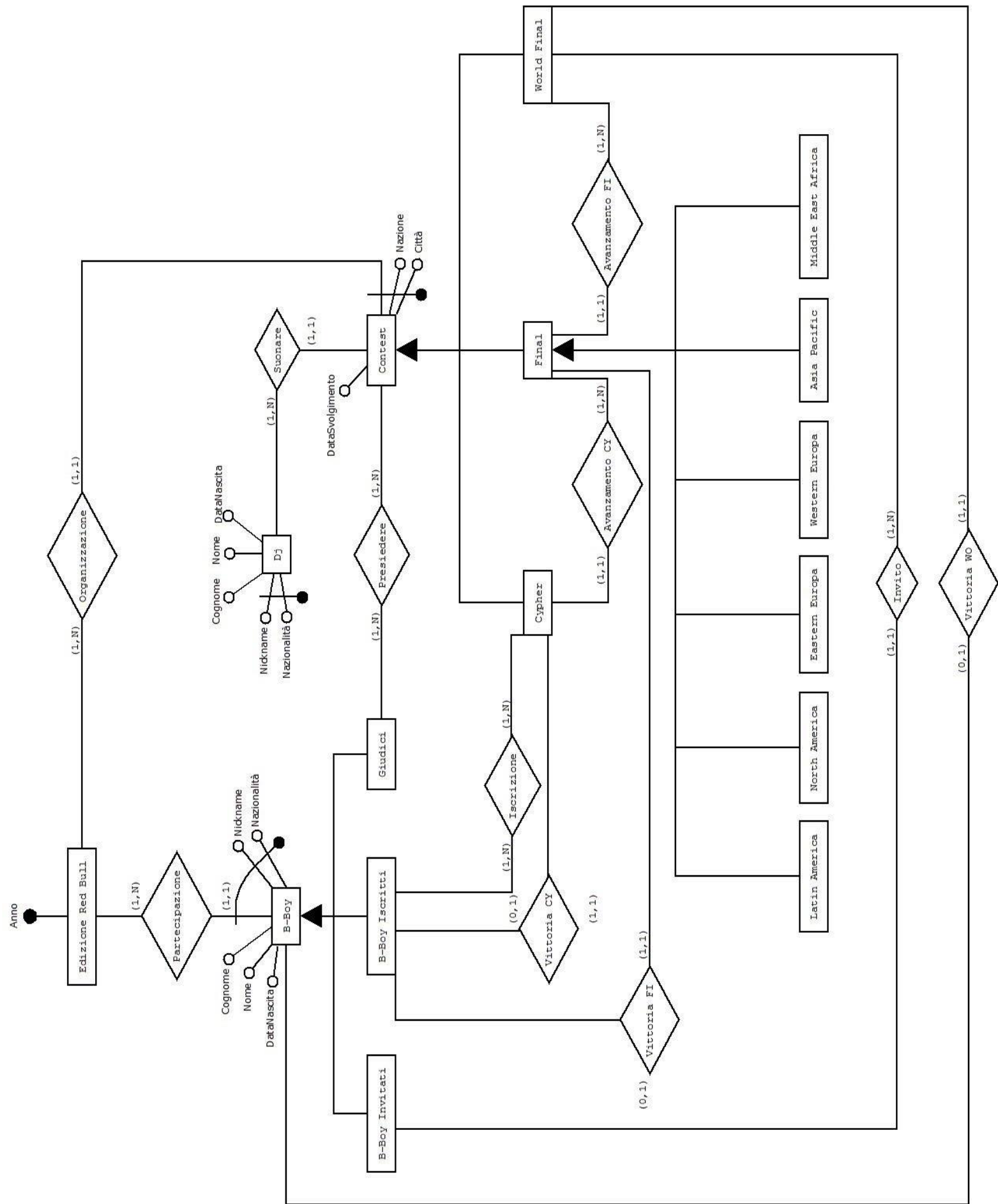
BBoy-WorldFinal: **VittoriaWorld**

- Solo un bboy (bboy iscritto o bboy invitato) *vince* un solo world final (0,1)
- Al world final *vince* uno ed un solo bboy (bboy iscritto o bboy invitato) (1,1)

3.3 Descrizione dei Vincoli di Integrità aggiuntivi

-vincolo tra **BBoyscritto-Cypher**: i contest Cypher sono svolti in una nazione specifica, solo BBoy di quella nazione si potranno iscrivere a quel contest. La nazionalità di un BBoy e la Nazione di svolgimento del Cypher devono essere la stessa.

4. Schema Entity - Relationship (E-R)



5. Ristrutturazione schema E-R

5.1 Eliminazione delle Generalizzazioni

- Generalizzazione su B-Boy

Elimino la generalizzazione su B-Boy accorpendo l'entità padre (B-Boy) alle entità figlie (BBoyInvitati / BBoyscritti / Giudici), per cui tutti gli attributi dell'entità B-Boy andranno sulle entità figlie.

Scelgo questa soluzione di eliminazione perché gli accessi che dovrò effettuare nelle entità figlie sono distinti, e mi serve sapere spesso quali BBoy hanno vinto o partecipato a qualche contest.

- Generalizzazione su Contest

Elimino la generalizzazione su Contest accorpendo l'entità padre (Contest) alle entità figlie (Cypher / Final / WorldFinal), per cui tutti gli attributi dell'entità Contest andranno sulle entità figlie.

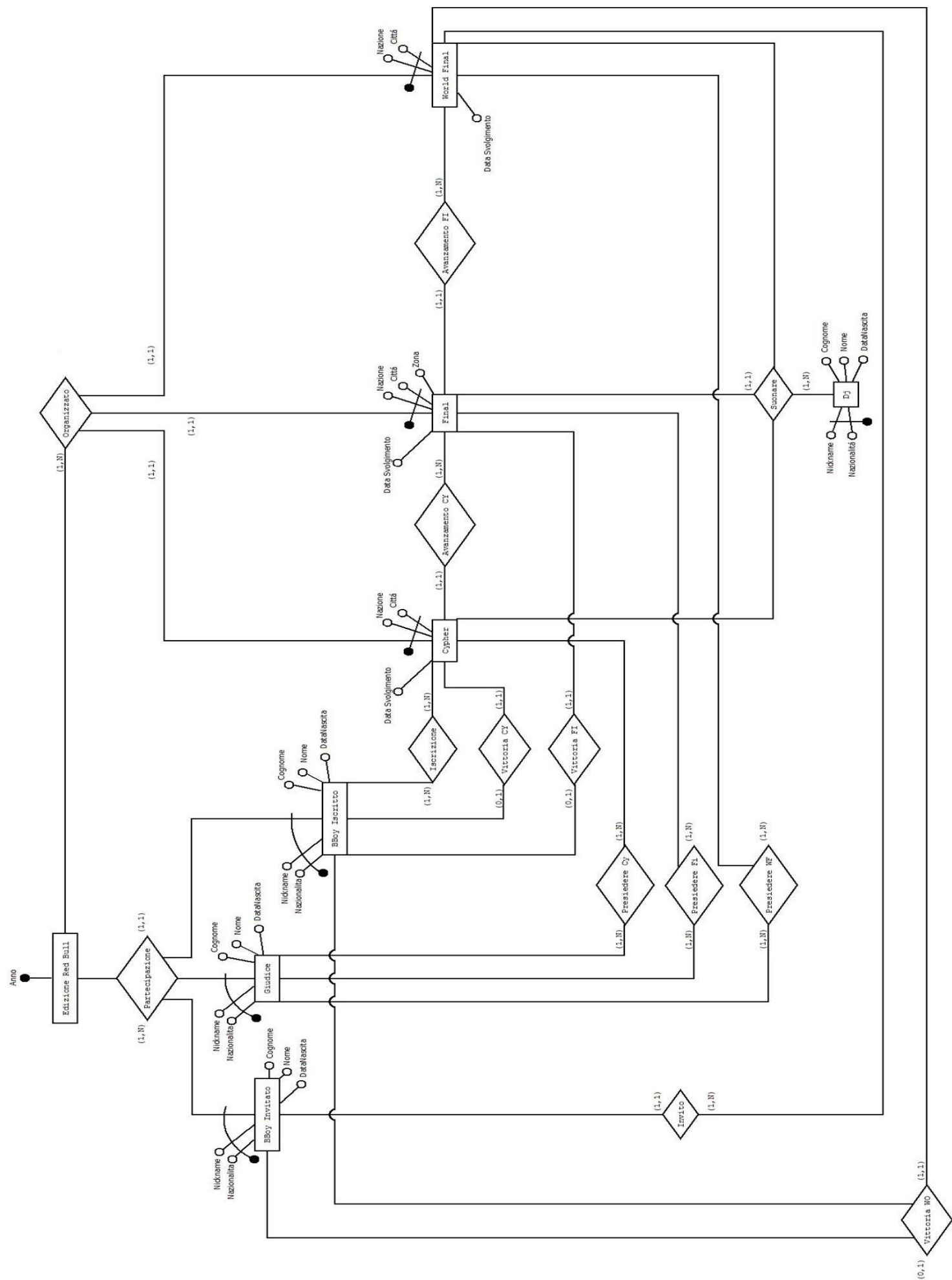
Scelgo questa soluzione di eliminazione perché gli accessi che dovrò effettuare nelle entità figlie sono distinti. In più scelgo questa soluzione per velocizzare la ricerca delle query, in quanto così facendo quando andrò a fare una query su un contest (Cypher, Final, WorldFinal) dovrò esaminare solo la tipologia di contest desiderata, senza dover scartare gli altri.

- Generalizzazione su Final

Elimino la generalizzazione su Final accorpendo le entità figlie (LatinAmerica / NorthAmerica / EasternEuropa / WesternEuropa / AsiaPacific / MiddleEastAfrica) all'entità padre (Final), per cui aggiungerò l'attributo "zona" all'entità, per indicare in quale zona verrà svolto il contest Final.

Scelgo questa soluzione di eliminazione in quanto gli accessi che dovrò effettuare all'entità padre sono contestuali a quelli delle figlie.

6. Schema Entity - Relationship (E-R) Ristrutturato



7. Progettazione Logica

PK: Primary Key

FK: Foreign Key

- EdizioneRedBull (Anno)
 - PK (Anno)
- BBoyInvitato (Nickname , Nazionalità , EdizioneRedBull , Cognome , Nome , DataNascita , NazioneWorld , CittàWorld)
 - PK (Nickname, Nazionalità, EdizioneRedBull)
 - EdizioneRedBull FK (EdizioneRedBull)
 - EdizioneRedBull, NazioneWorld, CittàWorld FK (WorldFinal)
- BBoyscritto (Nickname , Nazionalità , EdizioneRedBull , Cognome , Nome , DataNascita)
 - PK (Nickname , Nazionalità, EdizioneRedBull)
 - EdizioneRedBull FK (EdizioneRedBull)
- Giudice (Nickname , Nazionalità , EdizioneRedBull , Cognome , Nome , DataNascita)
 - PK (Nickname , Nazionalità, EdizioneRedBull)
 - EdizioneRedBull FK (EdizioneRedBull)
- Cypher (Nazione , Città , EdizioneRedBull , DataSvolgimento , NicknameBBoyVincitore NazionalitàBBoyVincitore , NazioneFinal , CittàFinal , NicknameDj , NazionalitàDj)
 - PK (Nazione, Città, EdizioneRedBull)
 - EdizioneRedBull FK (EdizioneRedBull)
 - NicknameBBoyVincitore NazionalitàBBoyVincitore, EdizioneRedBull FK (BBoyscritto)
 - NazioneFinal, CittàFinal, EdizioneRedBull FK (Final)
 - NicknameDj, NazionalitàDj FK (Dj)
- Final (Nazione , Città , EdizioneRedBull , DataSvolgimento , Zona , NicknameBBoyVincitore NazionalitàBBoyVincitore , NazioneWorldFinal , CittàWorldFinal , NicknameDj NazionalitàDj)
 - PK (Nazione, Città, EdizioneRedBull)
 - EdizioneRedBull FK (EdizioneRedBull)
 - NicknameBBoyVincitore, NazionalitàBBoyVincitore, EdizioneRedBull FK (BBoyscritto)
 - NazioneWorldFinal, CittàWorldFinal, EdizioneRedBull FK (WorldFinal)
 - NicknameDj, NazionalitàDj FK (Dj)
- WorldFinal (Nazione , Città , EdizioneRedBull , DataSvolgimento NicknameBBoyVincitoreInvitato , NazionalitàBBoyVincitoreInvitato NicknameBBoyVincitoreIscritto , NazionalitàBBoyVincitoreIscritto NicknameDj , NazionalitàDj)
 - PK (Nazione, Città, EdizioneRedBull)
 - NicknameDj, NazionalitàDj FK (Dj)
 - NicknameBBoyVincitoreInvitato, NazionalitàBBoyVincitoreInvitato, EdizioneRedBull FK (BBoyInvitato)
 - NicknameBBoyVincitoreIscritto, NazionalitàBBoyVincitoreIscritto, EdizioneRedBull FK (BBoyscritto)

- Dj (Nickname , Nazionalità , Cognome , Nome , DataNascita)
 - PK (Nickname, Nazionalità)
- Iscrizione (NicknameBBoylScritto , NazionalitàBBoylScritto , EdizioneRedBull NazioneCypher , CittàCypher)
 - PK (NicknameBBoylScritto , NazionalitàBBoylScritto , EdizioneRedBull , NazioneCypher , CittàCypher)
 - NicknameBBoylScritto, NazionalitàBBoylScritto, EdizioneRedBull *FK* (BBoylScritto)
 - NazioneCypher, CittàCypher, EdizioneRedBull *FK* (Cypher)
- PresiedereCypher (NicknameGiudice , NazionalitàGiudice , EdizioneRedBull NazioneCypher , CittàCypher)
 - PK (NicknameGiudice , NazionalitàGiudice , EdizioneRedBull , NazioneCypher , CittàCypher)
 - NicknameGiudice, NazionalitàGiudice, EdizioneRedBull *FK* (Giudice)
 - NazioneCypher, CittàCypher, EdizioneRedBull *FK* (Cypher)
- PresiedereFinal (NicknameGiudice , NazionalitàGiudice , EdizioneRedBull NazioneFinal , CittàFinal)
 - PK (NicknameGiudice , NazionalitàGiudice , EdizioneRedBull , NazioneFinal , CittàFinal)
 - NicknameGiudice, NazionalitàGiudice, EdizioneRedBull *FK* (Giudice)
 - NazioneFinal, CittàFinal, EdizioneRedBull *FK* (Final)
- PresiedereWorldFinal (NicknameGiudice , NazionalitàGiudice , EdizioneRedBull NazioneWorldFinal , CittàWorldFinal)
 - PK (NicknameGiudice , NazionalitàGiudice , EdizioneRedBull , NazioneWorldFinal , CittàWorldFinal)
 - NicknameGiudice, NazionalitàGiudice, EdizioneRedBull *FK* (Giudice)
 - NazioneWorldFinal, CittàWorldFinal, EdizioneRedBull *FK* (WorldFinal)
- Login (Username , Password , email)
 - PK (Username, Password)

8 Definizione dello schema logico tramite DDL di MySql

```
CREATE TABLE EdizioneRedBull (
```

```
    Anno          YEAR(4),  
    PRIMARY KEY (Anno)
```

```
)ENGINE = InnoDB;
```

```
CREATE TABLE BBoyInvitato (
```

```
    Nickname          VARCHAR(20),  
    Nazionalità        VARCHAR(20),  
    EdizioneRedBull   YEAR(4),  
    Cognome            VARCHAR(20),  
    Nome               VARCHAR(20),  
    DataNascita        DATE,  
    NazioneWorld       VARCHAR(20),  
    CittàWorld         VARCHAR(20),
```

```
    PRIMARY KEY (Nickname, Nazionalità, EdizioneRedBull),  
    FOREIGN KEY (EdizioneRedBull)  
        REFERENCES EdizioneRedBull (Anno),  
    FOREIGN KEY (NazioneWorld, CittàWorld, EdizioneRedBull)  
        REFERENCES WorldFinal (Nazione, Città, EdizioneRedBull)
```

```
)ENGINE = InnoDB;
```

```
CREATE TABLE BBoysIscritto (
```

```
    Nickname          VARCHAR(20),  
    Nazionalità        VARCHAR(20),  
    EdizioneRedBull   YEAR(4),  
    Cognome            VARCHAR(20),  
    Nome               VARCHAR(20),  
    DataNascita        DATE,
```

```
    PRIMARY KEY (Nickname, Nazionalità, EdizioneRedBull),  
    FOREIGN KEY (EdizioneRedBull)  
        REFERENCES EdizioneRedBull (Anno)
```

```
)ENGINE = InnoDB;
```

```
CREATE TABLE Giudice (
```

```
    Nickname          VARCHAR(20),  
    Nazionalità        VARCHAR(20),  
    EdizioneRedBull   YEAR(4),  
    Cognome            VARCHAR(20),  
    Nome               VARCHAR(20),  
    DataNascita        DATE,
```

```
    PRIMARY KEY (Nickname, Nazionalità, EdizioneRedBull),  
    FOREIGN KEY (EdizioneRedBull) REFERENCES EdizioneRedBull (Anno)
```

```
)ENGINE = InnoDB;
```

CREATE TABLE Cypher (

| | |
|--------------------------|--------------|
| Nazione | VARCHAR(20), |
| Città | VARCHAR(20), |
| EdizioneRedBull | YEAR(4), |
| DataSvolgimento | DATE, |
| NicknameBBoyVincitore | VARCHAR(20), |
| NazionalitàBBoyVincitore | VARCHAR(20), |
| NazioneFinal | VARCHAR(20), |
| CittàFinal | VARCHAR(20), |
| NicknameDj | VARCHAR(20), |
| NazionalitàDj | VARCHAR(20), |

PRIMARY KEY (Nazione, Città, EdizioneRedBull),
FOREIGN KEY (EdizioneRedBull)
REFERENCES EdizioneRedBull (Anno),
FOREIGN KEY (NicknameBBoyVincitore NazionalitàBBoyVincitore, EdizioneRedBull)
REFERENCES BBoyscritto (Nickname, Nazionalità, EdizioneRedBull),
FOREIGN KEY (NazioneFinal, CittàFinal, EdizioneRedBull)
REFERENCES Final (Nazione, Città, EdizioneRedBull),
FOREIGN KEY (NicknameDj, NazionalitàDj)
REFERENCES Dj (Nickname, Nazione)

)ENGINE = InnoDB;

CREATE TABLE Final (

| | |
|--------------------------|--------------|
| Nazione | VARCHAR(20), |
| Città | VARCHAR(20), |
| EdizioneRedBull | YEAR(4), |
| DataSvolgimento | DATE, |
| Zona | VARCHAR(20), |
| NicknameBBoyVincitore | VARCHAR(20), |
| NazionalitàBBoyVincitore | VARCHAR(20), |
| NazioneWorldFinal | VARCHAR(20), |
| CittàWorldFinal | VARCHAR(20), |
| NicknameDj | VARCHAR(20), |
| NazionalitàDj | VARCHAR(20), |

PRIMARY KEY (Nazione, Città, EdizioneRedBull),
FOREIGN KEY (EdizioneRedBull)
REFERENCES EdizioneRedBull (Anno),
FOREIGN KEY (NicknameBBoyVincitore NazionalitàBBoyVincitore, EdizioneRedBull)
REFERENCES BBoyscritto (Nickname, Nazionalità, EdizioneRedBull),
FOREIGN KEY (NazioneWorldFinal, CittàWorldFinal, EdizioneRedBull)
REFERENCES WorldFinal (Nazione, Città, EdizioneRedBull),
FOREIGN KEY (NicknameDj, NazionalitàDj)
REFERENCES Dj (Nickname, Nazione)

)ENGINE = InnoDB;

CREATE TABLE WorldFinal (

| | |
|----------------------------------|--------------|
| Nazione | VARCHAR(20), |
| Città | VARCHAR(20), |
| EdizioneRedBull | YEAR(4), |
| DataSvolgimento | DATE, |
| NicknameBBoyVincitoreInvitato | VARCHAR(20), |
| NazionalitàBBoyVincitoreInvitato | VARCHAR(20), |
| NicknameBBoyVincitoreIscritto | VARCHAR(20), |
| NazionalitàBBoyVincitoreIscritto | VARCHAR(20), |
| NicknameDj | VARCHAR(20), |
| NazionalitàDj | VARCHAR(20), |

PRIMARY KEY (Nazione, Città, EdizioneRedBull),
FOREIGN KEY (EdizioneRedBull)
REFERENCES EdizioneRedBull (Anno),
FOREIGN KEY (NicknameDj, NazionalitàDj)
REFERENCES Dj (Nickname, Nazione),
FOREIGN KEY (NicknameBBoyVincitoreInvitato, NazionalitàBBoyVincitoreInvitato,
EdizioneRedBull)
REFERENCES BBoyInvitato (Nickname, Nazionalità, EdizioneRedBull),
FOREIGN KEY (NicknameBBoyVincitoreIscritto, NazionalitàBBoyVincitoreIscritto,
EdizioneRedBull)
REFERENCES BBoylIscritto (Nickname, Nazionalità, EdizioneRedBull)

)ENGINE = InnoDB;

CREATE TABLE Dj (

| | |
|-------------|--------------|
| Nickname | VARCHAR(20), |
| Nazionalità | VARCHAR(20), |
| Cognome | VARCHAR(20), |
| Nome | VARCHAR(20), |
| DataNascita | DATE, |

PRIMARY KEY (Nickname, Nazionalità)

)ENGINE = InnoDB;

CREATE TABLE Iscrizione (

| | |
|--------------------------|--------------|
| NicknameBBoylIscritto | VARCHAR(20), |
| NazionalitàBBoylIscritto | VARCHAR(20), |
| EdizioneRedBull | YEAR(4), |
| NazioneCypher | VARCHAR(20), |
| CittàCypher | VARCHAR(20), |

PRIMARI KEY (NicknameBBoylIscritto, NazionalitàBBoylIscritto, EdizioneRedBull,
NazioneCypher, CittàCypher),
FOREIGN KEY (NicknameBBoylIscritto, NazionalitàBBoylIscritto, EdizioneRedBull)
REFERENCES BBoylIscritto (Nickname, Nazionalità, EdizioneRedBull),
FOREIGN KEY (NazioneCypher, CittàCypher, EdizioneRedBull)
REFERENCES Cypher (Nazione, Città, EdizioneRedBull)

)ENGINE = InnoDB;

```

CREATE TABLE PresiedereCypher (

    NicknameGiudice          VARCHAR(20),
    NazionalitàGiudice        VARCHAR(20),
    EdizioneRedBull          YEAR(4),
    NazioneCypher            VARCHAR(20),
    CittàCypher              VARCHAR(20),

    PRIMARI KEY (NicknameGiudice, NazionalitàGiudice, EdizioneRedBull, NazioneCypher,
                  CittàCypher),
    FOREIGN KEY (NicknameGiudice, NazionalitàGiudice, EdizioneRedBull)
        REFERENCES Giudice (Nickname, Nazionalità, EdizioneRedBull),
    FOREIGN KEY (NazioneCypher, CittàCypher, EdizioneRedBull)
        REFERENCES Cypher (Nazione, Città, EdizioneRedBull)

)ENGINE = InnoDB;

CREATE TABLE PresiedereFinal (

    NicknameGiudice          VARCHAR(20),
    NazionalitàGiudice        VARCHAR(20),
    EdizioneRedBull          YEAR(4),
    NazioneFinal             VARCHAR(20),
    CittàFinal               VARCHAR(20),

    PRIMARI KEY (NicknameGiudice, NazionalitàGiudice, EdizioneRedBull, NazioneFinal,
                  CittàFinal),
    FOREIGN KEY (NicknameGiudice, NazionalitàGiudice, EdizioneRedBull)
        REFERENCES Giudice (Nickname, Nazionalità, EdizioneRedBull),
    FOREIGN KEY (NazioneFinal, CittàFinal, EdizioneRedBull)
        REFERENCES Final (Nazione, Città, EdizioneRedBull),

)ENGINE = InnoDB;

CREATE TABLE PresiedereWorldFinal (

    NicknameGiudice          VARCHAR(20),
    NazionalitàGiudice        VARCHAR(20),
    EdizioneRedBull          YEAR(4),
    NazioneWorldFinal        VARCHAR(20),
    CittàWorldFinal          VARCHAR(20),

    PRIMARI KEY (NicknameGiudice, NazionalitàGiudice, EdizioneRedBull,
                  NazioneWorldFinal , CittàWorldFinal),
    FOREIGN KEY (NicknameGiudice, NazionalitàGiudice, EdizioneRedBull)
        REFERENCES Giudice (Nickname, Nazionalità, EdizioneRedBull),
    FOREIGN KEY (NazioneWorldFinal, CittàWorldFinal, EdizioneRedBull)
        REFERENCES WorldFinal (Nazione, Città, EdizioneRedBull)

)ENGINE = InnoDB;

```



```
CREATE TABLE Login (
```

```
    Username          VARCHAR(20),
    Password          VARCHAR(20),
    email             VARCHAR(20),
    PRIMARY KEY (Username, Password)
```

```
)ENGINE = InnoDB;
```

9 Query - Trigger - Funzioni

A seguito verranno riportare alcune query, trigger e funzioni significative utilizzate nell'interfaccia grafica per effettuare ricerche particolari oppure per visualizzare dati di interesse statistico

9.1 Query

9.1.1 Query

che visualizzi nickname di tutti i bboy che hanno vinto più di un world final, e mostrare anche in che anno lo hanno vinto.

```
SELECT WF1.NicknameBBoyVincitoreInvitato, WF1.EdizioneRedBull
FROM   WorldFinal WF1 JOIN WorldFinal WF2
      ON (WF1.NicknameBBoyVincitoreInvitato = WF2.NicknameBBoyVincitoreInvitato AND
          WF1.EdizioneRedBull <> WF2.EdizioneRedBull)
UNION
SELECT WF1.NicknameBBoyVincitoreIscritto, WF1.EdizioneRedBull
FROM   WorldFinal WF1 JOIN WorldFinal WF2
      ON (WF1.NicknameBBoyVincitoreIscritto = WF2.NicknameBBoyVincitoreIscritto AND
          WF1.EdizioneRedBull <> WF2.EdizioneRedBull)
```

| Nickname BBoy | Edizione RedBull |
|---------------|------------------|
| Hong 10 | 2006 |
| Hong 10 | 2014 |

9.1.2 Query

che visualizzi tutti i bboy che hanno vinto un final nel 2014 e fanno da giudice nel 2015, visualizzare anche in quale cypher presiedono come giudici.

```
SELECT Cypher.NicknameBBoyVincitore, Cypher.Nazione, Cypher.Citta
FROM   Cypher, Giudice, PresiedereCypher
WHERE  Cypher.EdizioneRedBull = "2014"
      AND Giudice.EdizioneRedBull = "2015"
      AND Cypher.NicknameBBoyVincitore = Giudice.Nickname
      AND PresiedereCypher.EdizioneRedBull = "2015"
      AND PresiedereCypher.NicknameGiudice = Giudice.Nickname
GROUP BY Cypher.NicknameBBoyVincitore
```

| Nickname BBoy | Nazione Cypher | Citta' Cypher |
|---------------|----------------|---------------|
| Blond | Australia | Singapore |
| Cheerito | Russia | Mosca |

9.1.3 Query

che conta quanti bboy si sono iscritti nel 2014 che hanno più di 25 anni e che nel 2015 sono stati invitati al world final.

```
SELECT COUNT(*) AS numeroBBoy
FROM   Iscrizione, BBoyscritto, BBoyInvitato
WHERE  Iscrizione.NicknameBBoyscritto = BBoyscritto.Nickname
      AND YEAR(CURDATE())-YEAR(BBoyscritto.DataNascita)>"25"
      AND BBoyInvitato.EdizioneRedBull = "2015"
      AND Iscrizione.NicknameBBoyscritto = BBoyInvitato.Nickname
```

| Numero BBoy |
|--------------------|
| 3 |

9.1.4 Query

che visualizza nazione, città e data svolgimento di un final nel 2014, il quale è svolto nella stessa nazione di un cypher del 2014, dove se ne sono svolti almeno 2

```
SELECT DISTINCT Final.Nazione, Final.Citta, Final.DataSvolgimento
FROM   Final, Cypher
WHERE  Final.EdizioneRedBull ="2014"
      AND Final.Nazione = any (SELECT c1.Nazione
                              FROM Cypher c1
                              WHERE c1.EdizioneRedBull = "2014"
                              GROUP BY c1.Nazione
                              HAVING COUNT(c1.Nazione) > "1")
```

| Nazione Final | Citta' Final | Data Svolgimento |
|----------------------|---------------------|-------------------------|
| America | Las Vegas | 2014-08-15 |

9.1.5 Query

che visualizza tutte le gare presiedute da quel giudice, e ordinare i giudici per nickname

```
SELECT *
FROM (SELECT Giudice.Nickname, Giudice.Nazionalita, Cypher.Nazione, Cypher.Citta
      FROM   Giudice, Cypher, PresiedereCypher
      WHERE  Giudice.Nickname = PresiedereCypher.NicknameGiudice
            AND PresiedereCypher.NazioneCypher = Cypher.Nazione
      UNION
      SELECT Giudice.Nickname, Giudice.Nazionalita, Final.Nazione, Final.Citta
      FROM   Giudice, Final, PresiedereFinal
      WHERE  Giudice.Nickname = PresiedereFinal.NicknameGiudice
            AND PresiedereFinal.NazioneFinal = Final.Nazione
      UNION
      SELECT Giudice.Nickname, Giudice.Nazionalita, WorldFinal.Nazione, WorldFinal.Citta
      FROM   Giudice, WorldFinal, PresiedereWorldFinal
      WHERE  Giudice.Nickname = PresiedereWorldFinal.NicknameGiudice
            AND PresiedereWorldFinal.NazioneWorldFinal = WorldFinal.Nazione
      ) as TAB
ORDER BY TAB.Nickname
```

| Nickname Giudice | Nazionalita' Giudice | Nazione | Citta' |
|-------------------------|-----------------------------|----------------|---------------|
| Admiracles | Belgio | Bulgaria | Sofia |
| Airdit | Germania | Romania | Bucarest |
| Airdit | Germania | Europa | Moldova |
| Andrezinho | Brasile | America | San Paolo |
| Aranha | Brasile | America | San Paolo |
| Beatmaster | Russia | Austria | Vienna |
| Benny | South Africa | South Africa | Johannesburg |
| Bigiki | Cina | Cina | Hong Kong |
| Blond | Australia | Australia | Singapore |
| Bootuz | Russia | Uzbekistan | Tashkent |
| Bootuz | Russia | Kyrgyzstan | Bishkek |
| Bootuz | Russia | Arzerbaijan | Baku |
| Bootuz | Russia | Georgia | Tbilisi |
| Bootuz | Russia | Tajikistan | Dushanbe |
| CHB | Egitto | Egitto | Cairo |
| Cheerito | Russo | Russia | Mosca |
| Chey | Spagna | Spagna | Madrid |

9.1.6 Query

che visualizza con quale frequenza, in percentuale, vengono effettuati i cypher durante l'anno 2014

```
SELECT MONTH(Cypher.DataSvolgimento) as Mesi,  
        COUNT(MONTH(Cypher.DataSvolgimento))/51  
FROM   Cypher  
GROUP BY Mesi
```

| Mese | Frequenza Cypher |
|-----------|------------------|
| Gennaio | 0 |
| Febbraio | 0.0333% |
| Marzo | 0.1333% |
| Aprile | 0.2667% |
| Maggio | 0.3833% |
| Giugno | 0.1500% |
| Luglio | 0 |
| Agosto | 0 |
| Settembre | 0 |
| Ottobre | 0 |
| Novembre | 0 |
| Dicembre | 0 |

9.2 Trigger

9.2.1 Trigger

che dopo aver inserito il contest world final iscrive al sito il BBoy vincitore, se non già iscritto. Avrà come Username e Password il suo Nickname

```
DELIMITER I  
CREATE TRIGGER creaUserSito  
AFTER INSERT ON WorldFinal  
FOR EACH ROW  
BEGIN  
    IF NOT EXISTS (SELECT L.Username  
                   FROM Login L  
                   WHERE L.Username = new.NicknameBBoyVincitoreInvitato  
                        OR L.Username = new.NicknameBBoyVincitoreIscritto) THEN  
        IF (new.NicknameBBoyVincitoreInvitato IS NOT NULL) THEN  
            INSERT INTO Login (Username, Password, email) VALUES  
            (new.NicknameBBoyVincitoreInvitato,new.NicknameBBoyVincitoreInvitato,  
            'email@email.com');  
        ELSE  
            INSERT INTO Login (Username, Password, email) VALUES  
            (new.NicknameBBoyVincitoreIscritto,new.NicknameBBoyVincitoreIscritto ,  
            'email@email.com');  
        END IF;  
    END IF;  
END I  
DELIMITER ;
```

9.2.2 Trigger

che controlla se l'e-mail è stata inserita correttamente, per cui nel formato "_ @ _ . _"

```
DELIMITER I
CREATE TRIGGER controllaMail
BEFORE INSERT ON Login
FOR EACH ROW
BEGIN
    DECLARE errore VARCHAR(20);
    IF(new.email NOT LIKE '%_@%_._%') THEN
        SELECT Username INTO errore
        FROM Login
        LIMIT 1;
        INSERT INTO Login (Username) VALUES (errore);
    END IF;
END I
DELIMITER ;
```

9.3 Funzioni

9.3.1 Funzione

che restituisce il numero di vittorie complessive di un certo BBoy

```
DELIMITER I
CREATE FUNCTION numeroVittorie (NicknameBBoy VARCHAR(20))
RETURNS INT
BEGIN
    DECLARE numeroCY INT;
    DECLARE numeroFI INT;
    DECLARE numeroWF INT;
    SELECT COUNT(*) INTO numeroCY
    FROM Cypher
    WHERE NicknameBBoyVincitore = NicknameBBoy ;
    SELECT COUNT(*) INTO numeroFI
    FROM Final
    WHERE NicknameBBoyVincitore = NicknameBBoy ;
    SELECT COUNT(*) INTO numeroWF
    FROM WorldFinal
    WHERE NicknameBBoyVincitoreInvitato = NicknameBBoy
        OR NicknameBBoyVincitoreIscritto = NicknameBBoy ;
    RETURN numeroCY+numeroFI+numeroWF ;
END I
DELIMITER ;
```

9.3.2 Funzione

che restituisce il Nickname del BBoy (Invitato, Iscritto o Giudice) più vecchio in un determinato anno

```
DELIMITER I
CREATE FUNCTION piuVecchio(annoRedBull YEAR(4))
RETURNS VARCHAR(500)
BEGIN
    DECLARE oldInv VARCHAR(20);
    DECLARE dataInv DATE;
    DECLARE oldIsc VARCHAR(20);
    DECLARE dataIsc DATE;
    DECLARE oldGiu VARCHAR(20);
    DECLARE dataGiu DATE;
    SELECT Nickname, DataNascita INTO oldInv, dataInv
    FROM BBoyInvitato
    WHERE DataNascita <= All (SELECT DataNascita FROM BBoyInvitato)
        AND EdizioneRedBull = annoRedBull;
    SELECT Nickname, DataNascita INTO oldIsc, dataIsc
    FROM BBoylscritto
    WHERE DataNascita <= All (SELECT DataNascita FROM BBoylscritto)
        AND EdizioneRedBull = annoRedBull;
    SELECT Nickname, DataNascita INTO oldGiu, dataGiu
    FROM Giudice
    WHERE DataNascita <= All (SELECT DataNascita FROM Giudice)
        AND EdizioneRedBull = annoRedBull;
    IF (dataInv < dataIsc) THEN
        IF (dataInv < dataGiu) THEN
            RETURN oldInv;
        END IF;
    ELSEIF (dataIsc < dataGiu) THEN
        RETURN oldIsc;
    ELSE
        RETURN oldGiu;
    END IF;
END I
DELIMITER;
```

10 Interfaccia WEB

Una funzione importante è la funzione di login e di registrazione, in quanto solo se registrati al sito si possono effettuare query sul database. Quando l'utente effettua il login la pagina dinamica verifica se Username e Password coincidono con una entry della tabella Login. Nel caso di risposta affermativa l'utente viene ridirezionato verso la sua homepage per effettuare le query. In caso non si possieda un Username e Password bisognerà registrarsi, essendo così inviati al form di registrazione al sito. Per questo sistema di login è stata utilizzata la tecnica delle *sessioni*. Una volta effettuato il login:

- se utente "Amministratore" si verrà indirizzati alla pagina *homeAmministratore.php* e si avranno questi privilegi:
 - Eseguire query
 - Visualizzare Query - Trigger - Funzioni del progetto
 - Visualizzare tutti gli Utenti registrati al sito
 - Inserire nuovi Utenti nel sito
 - Modificare Utenti già registrati nel sito
 - Cancellare Utenti già registrati nel sito
- se altro utente si verrà indirizzati alla pagina *homeQuery.php* e si avranno questi privilegi:
 - Eseguire query
 - Visualizzare Query - Trigger - Funzioni del progetto

Ora andiamo a visualizzare le funzioni delle varie pagine.

10.1 home.php

La pagina *home.php* è solamente la pagina iniziale dove darà la possibilità tramite un pulsante di collegarsi alla pagina di *login.php*

10.2 login.php

Nella pagina *login.php* bisognerà inserire Username e Password già esistenti, per poi essere reindirizzati alla pagina *homeQuery.php* dove si potranno effettuare le vari query. Se non si è in possesso di un Username e una Password bisognerà schiacciare nella scritta "Registrati" per essere reindirizzati alla pagina di registrazione. Per comodità nel testare il progetto ho fatto visualizzare gli username e le password già inserite.

10.3 verificaLogin.php

La pagina *verificaLogin.php* serve solamente per verificare se l'Username e la Password inserita nella pagina *login.php* corrisponde ad una entry nella tabella "Login" del database. Se l'Username e la Password inserite sono corrette, verrà iniziata una sessione che avrà come variabili di sessione l'Username e la Password, e si verrà reindirizzati alla pagina *homeQuery.php* per poter iniziare a visualizzare e provare le varie query. Invece se l'Username e la Password inserite sono errate si verrà reindirizzati alla pagina *login.php*

10.4 registrazione.php

La pagina *registrazione.php* consiste nel form per l'inserimento dei dati per poi effettuare il login al sito. Bisognerà inserire un Username una Password e un email valida (basterà seguire l'esempio nel form, altrimenti si attiverà il trigger proposto e non verrà effettuata la registrazione). Inviato il form di registrazione si verrà reindirizzati alla pagina di login, per poi effettuare il login al sito

10.5 logout.php

La pagina *logout.php* fa solamente il logout al sito, settando a null le variabili di sessione.

10.6 jQuery.js

La pagina *jQuery.js* è una libreria che permette di usare una funzione ajax. Questa libreria è inclusa nella pagina *homeQuery.php*

10.7 homeQuery

La pagina *homeQuery.php* si può considerare come la vera home, in quanto è la pagina che ti permette di effettuare query istantanee sul database scegliendo le tabelle tramite dei menu a tendina, e si possono scegliere anche quali campi della tabella visualizzare tramite dei checkbox. Per visualizzare i checkbox relativi ad una tabella (selezionata con il menu a tendina) senza dover ricaricare la pagina ho usato delle funzioni jQuery, ecco perché l'esistenza della libreria jQuery.js. Oltre alle query immediate, dalla pagina *homeQuery.php* ci si può collegare, tramite tre pulsanti, alle tre pagine relative al progetto per visualizzare le 6 query, i 2 trigger e le 2 funzioni presentate nel progetto

10.8 ajaxRequest.php

La pagina *ajaxRequest.php* non fa altro che elaborare il codice jQuery, creando e facendo visualizzare i checkbox relativi ad una certa tabella senza dover ricaricare la pagina *homeQuery.php*

10.9 queryProgetto.php

La pagina *queryProgetto.php* è la pagina dove vengono visualizzate e provate le 6 query proposte nel progetto, si potrà scegliere quale query provare tramite un menu a tendina.

10.10 triggerProgetto.php

La pagina *triggerProgetto.php* è la pagina dove vengono visualizzate e i 2 trigger proposti nel progetto, si potrà scegliere quale trigger visualizzare tramite un menu a tendina.

10.11 funzioniProgetto.php

La pagina *funzioniProgetto.php* è la pagina dove vengono visualizzate e provate le 2 funzioni proposte nel progetto, si potrà scegliere quale funzione provare tramite un menu a tendina.

10.12 homeAmministratore.php

La pagina *homeAmministratore.php* è la pagina che si può accedere solamente se si è in possesso dell'account amministratore (Username: "amministratore" ; Password: "amministratore"). Da questa pagina si avrà la possibilità di gestire tutti gli utenti registrati al sito, collegandosi tramite dei pulsanti si potrà visualizzare tutti gli utenti registrati, si potrà inserire nuovi utenti nel sito, si potrà modificare dei campi di un utente e in fine si potrà cancellare degli utenti dal sito. Tramite un pulsante si può ricollegarsi alla pagina *homeQuery.php* così da poter provare le stesse cose di un utente normale.

10.13 visualizza.php

La pagina *visualizza.php*, accessibile solo dell'amministratore, permette solo di visualizzare tutti i campi degli utenti registrati al sito.

10.14 inserisci.php

La pagina *inserisci.php*, accessibile solo dell'amministratore, permette tramite un form di inserire nuovi utenti al sito.

10.15 modifica.php

La pagina *modifica.php*, accessibile solo dell'amministratore, permette di selezionare l'username dell'utente da voler modificare, una volta selezionato si verrà reindirizzati alla pagina *modificaRecord.php* dove verrà effettuata la modifica.

10.16 modificaRecord.php

La pagina *modificaRecord.php*, accessibile solo dell'amministratore, permette la modifica di tutti i campi dell'utente selezionato nella pagina *modifica.php*

10.17 cancella.php

La pagina *cancella.php*, accessibile solo dell'amministratore, permette di cancellare un utente registrato al sito.