

# **PROGETTO**

# **BASI DI DATI**

## **Componenti del gruppo:**

- Corò Giulia  
username: gcor
- Pozza Marco  
username: mapozza

## **Account sul quale è stato caricato il progetto:**

- gcor

# **ABSTRACT**

Love Live: School Idol Festival (da ora in poi SIF), è un gioco per iOS e Android rilasciato in Giappone il 16 Aprile 2013, e nel resto del mondo il 12 Maggio 2014. Esso consiste nel formare dei Team di 9 carte, con i quali il giocatore andrà poi a giocare varie canzoni premendo con il giusto ritmo dei cerchietti che appariranno, ed in base al punteggio ottenuto, otterrà ricompense e/o nuove carte. Il punteggio dipenderà sia dall'abilità del giocatore sia dalla composizione del suo Team.

Le carte, raffiguranti varie School Idol (ossia ragazze popolari nel mondo dello spettacolo in vari settori - in questo caso nel canto e ballo), sono ad oggi circa 850 e continuano a crescere, con l'uscita ogni mese di nuove carte.

La base di dati che abbiamo realizzato servirà per un sito che raccoglie informazioni riguardanti SIF rivolte ai giocatori, i quali potranno compiere svariate operazioni di ricerca, quali:

- ➔ Cercare informazioni riguardanti una determinata carta, come la rappresentazione grafica, le informazioni riguardanti l'edizione, il set a cui appartiene (se esiste), se è ottenibile tramite evento o meno, e visualizzarne tutte le statistiche;
- ➔ Confrontare le statistiche di varie carte, per trovare quella più adatta da inserire nella propria squadra;
- ➔ Visualizzare un elenco delle ultime carte uscite nel mese o anno corrente, oppure delle carte uscite in un determinato mese;
- ➔ Vedere un elenco degli eventi passati, presenti e futuri del gioco, e le canzoni giocabili in ciascuno degli eventi;
- ➔ Guardare chi interpreta una determinata canzone del gioco, che sia una school idol solista, un gruppo o una subunit, inoltre vedere se la canzone fa parte di un album e quale;
- ➔ Visualizzare tutte le informazioni di una School Idol, come a quale scuola appartiene, quali canzoni interpreta, se fa parte di un gruppo o di una subunit;

## **DESCRIZIONE E ANALISI DEI REQUISITI**

Una carta di SIF, identificata univocamente da un ID, è caratterizzata da una rappresentazione grafica, una Skill (abilità), una Center Skill (abilità speciale che si attiva solo quando la carta è posta al centro del proprio team), un'Attributo il quale determina anche il colore della carta (Smile=Rosa, Pure=Verde, Cool=Azzurro), una rarità che può essere Normal, Rare, Super Rare o Ultra Rare (N, R, SR, UR), e da un livello massimo raggiungibile, il quale cambia appunto da carta a carta, a seconda della rarità di essa. Inoltre le carte hanno delle statistiche, ovvero dei valori, chiamati Smile, Pure, Cool e Stamina che aumentano con l'aumentare del livello, delle quali a noi interessa registrare solo i valori che raggiungono al livello massimo.

Le carte si dividono in carte giocabili e di supporto. Una carta giocabile può essere inserita nel proprio team, mentre una carta di supporto può soltanto essere usata per far salire di livello le altre carte. Queste ultime infatti non sono caratterizzate da un livello né da delle statistiche, ma soltanto da una Skill che determinerà quali effetti avranno quando usate per alzare di livello le altre carte.

Le carte giocabili inoltre si dividono in Base o Idolizzate: una carta base può essere "idolizzata" fondendola con un'altra carta uguale ad essa, una carta di questo tipo avrà una nuova rappresentazione grafica, livello massimo aumentato e di conseguenza anche tutte le statistiche aumentate. Le versioni base e idolizzata della stessa carta hanno l'ID composto dalla medesima parte numerica, ma seguito da 'b' nel caso siano la versione base, e da 'i' se idolizzate.

Una carta, sia giocabile che di supporto, può o meno appartenere ad un Set, ossia una raccolta di carte con rappresentazioni legate dalla stessa tematica. Un Set è identificato univocamente dal suo nome, ed ha una data di rilascio.

Ogni carta giocabile rappresenta una School Idol, ossia una adolescente divenuta popolare, in questo caso grazie al canto e al ballo. Ogni Idol, identificata da Nome e Cognome, ha una data di nascita e frequenta il primo, secondo o terzo anno di una determinata scuola.

Una School Idol appartiene ad uno e un solo gruppo, identificato dal nome e avente un preciso numero di membri; inoltre può appartenere ad una Subunit (ossia un sottogruppo formatasi all'interno di un gruppo), anch'esso identificato dal nome. Non è detto che tutti i gruppi abbiano Subunit, per cui una School idol può anche non appartenere a nessuna Subunit, ma soltanto al gruppo stesso. Un gruppo può avere più Subunit al suo interno, e una Subunit deve per forza aver un Gruppo di cui essa è sottogruppo. Ogni Gruppo e Subunit deve avere un minimo di 2 membri per essere definito tale.

Le canzoni giocabili, identificate dal Titolo e aventi un attributo (Smile, Pure o Cool), una durata e una data di uscita, sono solitamente cantate da un Gruppo o da una Subunit, ma possono essere anche cantate come solista da una School Idol.

Le canzoni possono o meno appartenere ad un Album che le raccoglie, il quale è identificato da un Titolo ed una Data di Rilascio.

In SIF sono organizzati periodicamente degli eventi. Questi sono identificati dal Nome e caratterizzati dalla tipologia di evento (Score Match, Medley, Token, Challenge Festival), da una data di inizio ed una di fine. Nel corso di un evento si possono giocare una determinata lista di canzoni, accumulando punti evento: al raggiungimento di un determinato punteggio può essere vinta una specifica carta premio. Gli eventi possono essere passati, attualmente in corso o futuri.

# **PROGETTAZIONE CONCETTUALE**

## **DESCRIZIONE ENTITA':**

### **CARTA**

Rappresentazione di una specifica carta presente nel gioco, con tutte le sue caratteristiche.

#### **Attributi:**

- ID: char(5)
- Rappresentazione grafica: BLOB
- Skill: CLOB
- Attributo: varchar(20)
- Rarità: varchar(2)

### **CARTA\_SUPPORTO**

Specializzazione dell'entità Carta, rappresenta una carta non giocabile direttamente, ma usata per alzare di livello le carte giocabili.

#### **Attributi:**

- Nome: varchar(20)
- Inoltre eredita tutti gli attributi dell'entità Carta

### **CARTA\_GIOCABILE**

Specializzazione dell'entità Carta, rappresenta una carta giocabile, ovvero che può essere inserita nel Team di gioco.

#### **Attributi:**

- Center skill: CLOB
- Livello max: tinyint
- Statistiche al Livello Max (attributo composto):
  - Stamina: tinyint
  - Smile: tinyint
  - Pure: tinyint
  - Cool: tinyint
- Inoltre eredita tutti gli attributi dell'entità Carta

### **BASE**

Specializzazione di Carta\_giocabile, rappresenta la carta nella sua versione base.

#### **Attributi:**

- Eredita tutti gli attributi di Carta\_giocabile

### **IDOLIZZATA**

Specializzazione di Carta\_giocabile, rappresenta la carta nella sua versione idolizzata, ossia un potenziamento di quella base, con statistiche aumentate e diversa rappresentazione grafica.

#### **Attributi:**

- Eredita tutti gli attributi di Carta\_giocabile

### **SET**

Alcune carte fanno parte di collezioni (chiamate Set), dove ogni idol ha un vestito particolare attinente ad una tematica specificata appunto dal nome del Set.

#### **Attributi:**

- Nome: varchar(20)
- Data Rilascio: date

### **SCHOOL\_IDOL**

Entità che rappresenta la ragazza idol fisica, quella che poi è rappresentata graficamente sulla carta.

#### **Attributi:**

- Nome varchar(20)
- Cognome: varchar(20)
- Data di nascita: date
- Scuola: varchar(20)
- Anno di frequenza: tinyint

### **GRUPPO**

Rappresenta i gruppi che le idol formano per poi esibirsi.

#### **Attributi:**

- Nome: varchar(20)
- Numero di componenti: tinyint

### **SUBUNIT**

Un sottoinsieme dell'entità gruppo. Un gruppo infatti può a sua volta essere formato da dei sottogruppi delle stesse idol che lo formano, questo per poter permettere esibizioni separate.

#### **Attributi:**

- Nome: varchar(20)

### **CANZONE**

Entità che rappresenta le canzoni che i vari gruppi di idol scrivono e interpretano nelle loro esibizioni.

#### **Attributi:**

- Titolo: varchar(20)
- Attributo: varchar(20)
- Durata: time
- Data rilascio: date

### **ALBUM**

Una raccolta di canzoni in uno stesso CD.

#### **Attributi:**

- Data Rilascio: date
- Titolo: varchar(20)

### **EVENTO**

Entità che rappresenta un evento, al quale si può partecipare in un determinato periodo per tentare di vincere una particolare carta rara.

#### **Attributi:**

- Nome: varchar
- Tipologia: varchar
- Data inizio: date
- Data fine: date

### **EVENTO PASSATO**

Specializzazione dell'entità Evento, indica un evento che si è già concluso.

#### **Attributi:**

-Eredita tutti gli attributi di Evento.

### **EVENTO PRESENTE**

Specializzazione dell'entità Evento, indica un evento che è in corso attualmente (iniziato ma non concluso).

#### **Attributi:**

-Eredita tutti gli attributi di Evento.

### **EVENTO FUTURO**

Specializzazione dell'entità Evento, indica un evento che è in programma ma che deve ancora cominciare.

#### **Attributi:**

-Eredita tutti gli attributi di Evento.

## DESCRIZIONE GERARCHIE:

- L'entità carte ha una gerarchia che descrive più in dettaglio le differenze tra varie tipologie di carte, questo perchè non tutte le carte son uguali e utilizzabili in gioco. I tipi di carte esistenti sono (generalizzazione totale) :
  - CARTE SUPPORTO: particolari carte che non possono essere utilizzate all'interno del team ma son usate per far salire di livello le altre CARTE GIOCABILI in quanto forniscono un significativo ammontare di esperienza quando vengono "sacrificate".
  - CARTE GIOCABILI: tutte le altre carte, che possono quindi essere usate per costruire il proprio team di gioco.
- Le CARTE GIOCABILI si suddividono in (generalizzazione totale):
  - BASE: le carte come vengono ottenute inizialmente. Esse possono in futuro diventare IDOLIZZATE, tramite la "fusione" della stessa carta due volte.
  - IDOLIZZATE: carte ottenibili dalla fusione di due carte BASE uguali. Il loro livello massimo aumenta, quindi anche tutte le statistiche. Gli effetti son generalmente potenziati.
- Gli EVENTI si suddividono in (generalizzazione totale):
  - EVENTI PASSATI: tutti gli eventi che si sono svolti nel gioco e si sono già conclusi.
  - EVENTI PRESENTI: ossia l'evento che è attualmente in corso.
  - EVENTI FUTURI: tutti gli eventi che sono in programma ma devono ancora iniziare.

## DESCRIZIONE RELAZIONI:

### 1) Carta – Set: **Appartiene**

- Una carta appartiene ad una unico set, o a nessuno.
- Un set può avere da 1 a N carte appartenenti ad esso.

### 2) Carta – School Idol: **Rappresentazione**

- Una carta rappresenta graficamente una school idol.
- Una school idol deve avere almeno una carta che la rappresenta per esistere all'interno del gioco, ma può averne di più.

### 3) School Idol – Gruppo: **Membri**

- Una school idol è membro di uno ed un solo gruppo. Non può non appartenere a nessun gruppo.
- Un gruppo ha come membri almeno 2 school idol, e può averne fino ad N.

### 4) Canzone – Gruppo/sottogruppo/idol : **Interpretazione**

- Una canzone può essere cantata da un gruppo, da una subunit oppure da idol soliste.
- Gruppi, subunit e idol possono creare e cantare più canzoni.

### 5) Album – Canzone: **Composizione**

- Un album è composto da almeno 2 canzoni (per cui singoli non sono intesi come album)
- Una canzone può esser parte di un solo album o di nessuno

### 6) Canzone – Evento: **Giocabile**

- Una canzone può esser resa giocabile all'interno di un evento
- In un evento sono giocabili da 1 a N canzoni.

### 7)Evento – Carta: **Vincita**

#### **Attributo:**

Punteggio\_vincita: int

Questo punteggio determina o meno la vincita della carta premio.

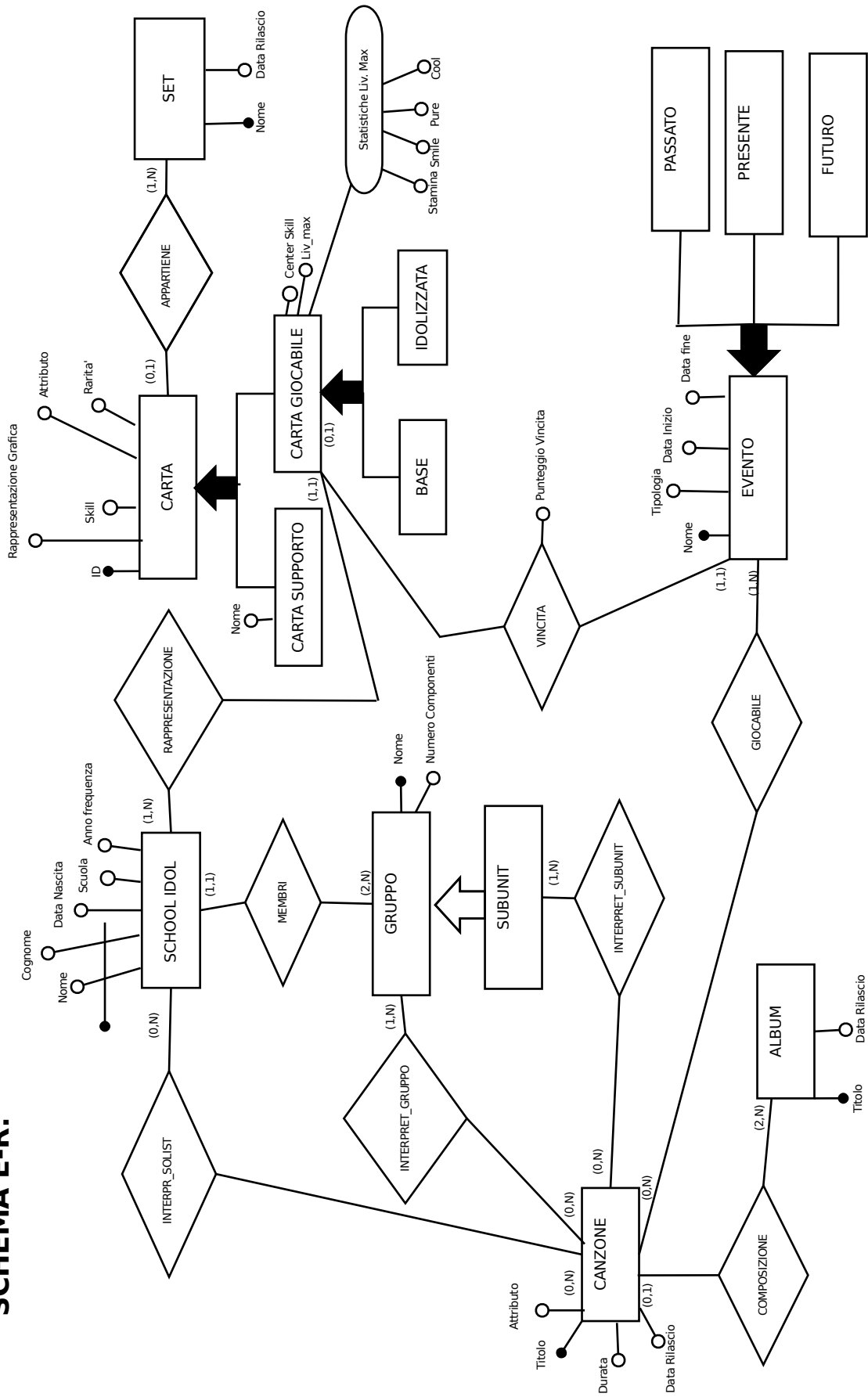
- In un evento può essere vinta una sola carta (al raggiungimento del punteggio necessario).
- Una carta può essere vinta in un unico evento, oppure in nessuno.

## VINCOLI D'INTEGRITA':

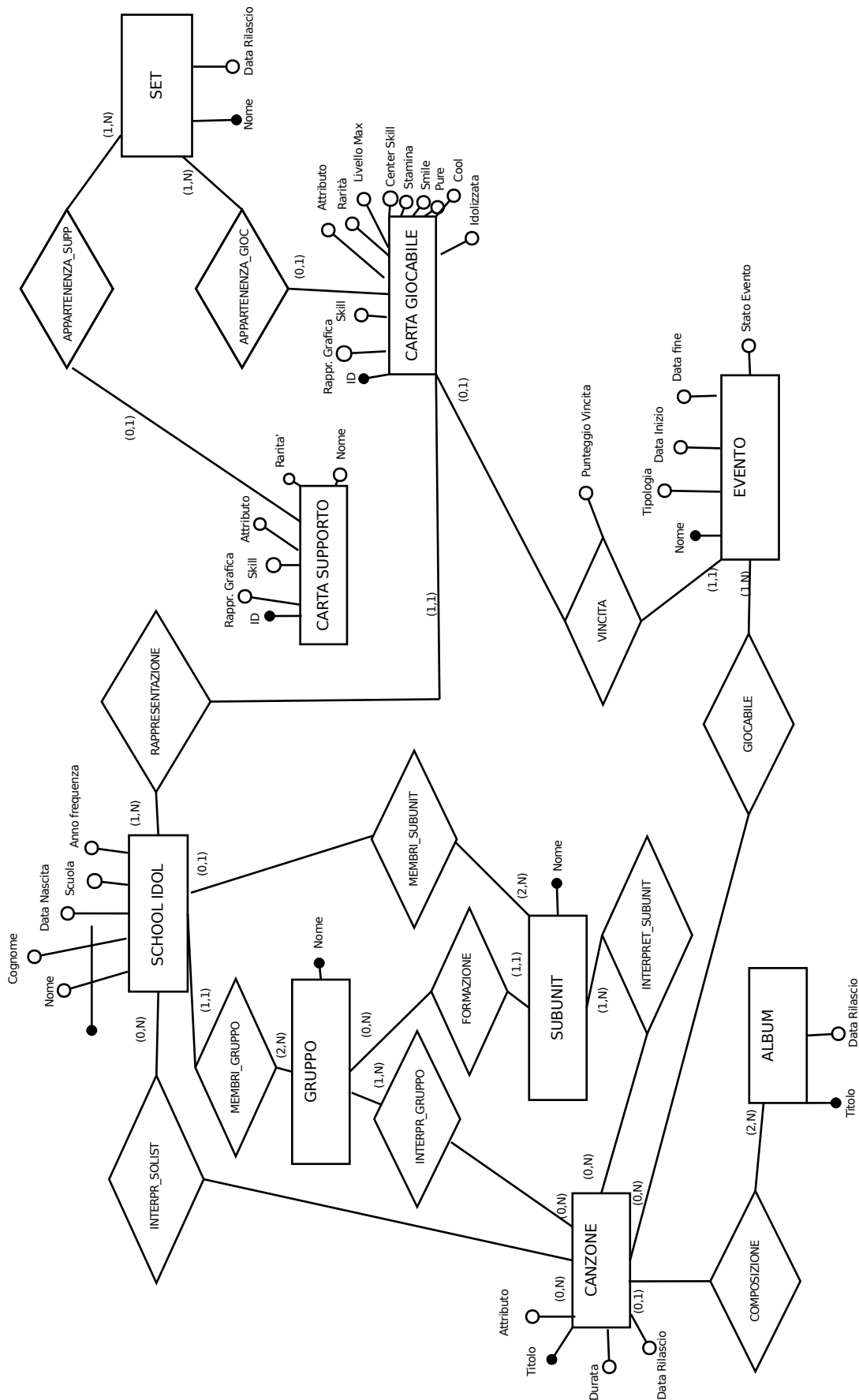
- Se una carta appartiene ad un set deve per forza esser rappresentata da una particolare idol
- Se una carta rappresenta una idol, quell'idol deve esistere
- Se un sottogruppo è formato da determinate idol, anche il gruppo totale deve contenere quelle idol
- Le canzoni di un gruppo non possono esser canzoni di altri gruppi, subunit o soliste
- Un gruppo o una subunit per esistere deve avere almeno 2 membri
- Un album deve essere composto da almeno 2 canzoni



# SCHEMA E-R:



SCHEMA E-R RISTRUTTURATO:



## **Attività di ristrutturazione e motivazione delle scelte:**

1. Analisi delle Ridondanze:  
Viene rimosso l'attributo 'Numero Componenti' dall'entità Gruppo poiché è un attributo derivabile in altro modo. Non ci è utile tenerlo poiché non è un'operazione eseguita frequentemente.
2. Eliminazione delle Generalizzazioni:
  - Viene accorpata l'entità Carta nelle entità figlie Carta\_supporto e Carta\_giocabile, poiché gli accessi alle due entità figlie sono solitamente distinti. Tutti gli attributi di Carta vengono quindi ereditati da ognuna delle due entità figlie.
  - Vengono accorpate le entità Base e Idolizzata nell'entità genitore Carta\_giocabile, dato che in questo caso gli accessi sono contestuali. Viene aggiunto quindi all'entità Carta\_giocabile un attributo 'Idolizzata' di tipo booleano che avrà valore false se la carta è Base, e true se Idolizzata.
  - La generalizzazione tra Gruppo e Subunit viene sostituita con una relationship, così da permettere accessi separati alle due entità figlia e genitore.
  - La generalizzazione di Evento in Passati, Presenti e Futuri viene accorpata nell'entità genitore Evento con l'aggiunta ad essa di un attributo varchar(20) 'Stato\_evento', il quale potrà avere come valori 'Passato', 'Presente', o 'Futuro'.
3. Partizionamento o accorpamento di entità e relationship:  
Nessuna modifica effettuata
4. Scelta degli identificatori principali:  
Tutti gli identificatori scelti in precedenza soddisfano i criteri di assenza di opzionalità, semplicità ed utilizzo delle operazioni più importanti, perciò non è necessario introdurre nuovi codici come identificatori.

# PROGETTAZIONE LOGICA

## **Schema logico:**

Carta\_giocabile(ID, Rappresentazione\_grafica, Idolizzata, Liv\_max, Skill, Attributo, Rarità, Center\_skill, Stamina, Smile, Pure, Cool, Set\_appartenenza, Nome\_idol, Cognome\_idol)

Carta\_supporto(ID, Rappresentazione\_grafica, Skill, Attributo, Rarità, Set\_appartenenza, Nome)

Set\_carta(Nome, Data\_rilascio)

School\_Idol(Nome, Cognome, Data\_nascita, Scuola, Anno\_frequenza, Gruppo, Subunit)  
Gruppo(Nome)

Subunit(Nome, Gruppo\_appartenenza)

Canzone(Titolo, Data\_rilascio, Durata, Attributo, Album)

Album(Titolo, Data\_rilascio)

Interpretazione\_solista(Nome\_idol\_solista, Cognome\_idol\_solista, Canzone)

Interpretazione\_gruppo(Gruppo, Canzone)

Interpretazione\_subunit(Subunit, Canzone)

Evento(Nome, Tipologia, Data\_inizio, Data\_fine, Stato\_evento, Carta\_premio, Punteggio\_vincita)

Canzoni\_evento(Evento, Canzone)

## **Vincoli di integrità:**

- V.i.r. tra Set\_appartenenza in Carta\_giocabile e la chiave di Set\_carta
- V.i.r. tra Set\_appartenenza in Carta\_supporto e la chiave di Set\_carta
- V.i.r. tra Nome\_idol, Cognome\_idol in Carta\_giocabile e la chiave di School\_Idol; valori nulli non ammessi
- V.i.r. tra Gruppo in School\_idol e la chiave di Gruppo, valori nulli non ammessi
- V.i.r. tra Subunit in School\_idol e la chiave di Subunit
- V.i.r. tra Gruppo\_appartenenza in Subunit e la chiave di Gruppo, valori nulli non ammessi
- V.i.r. tra Nome\_idol\_solista, Cognome\_idol\_solista in Interpretazione\_solista e la chiave di School\_idol
- V.i.r. tra Canzone in Interpretazione\_solista e la chiave di Canzone
- V.i.r. tra Gruppo in Interpretazione\_gruppo e la chiave di Gruppo
- V.i.r. tra Canzone in Interpretazione\_gruppo e la chiave di Canzone
- V.i.r. tra Subunit in Interpretazione\_subunit e la chiave di Subunit
- V.i.r. tra Canzone in Interpretazione\_subunit e la chiave di Canzone
- V.i.r. tra Album in Canzone e la chiave di Album
- V.i.r. tra Evento in Canzoni\_evento e la chiave di Evento
- V.i.r. tra Canzone in Canzoni\_evento e la chiave di Canzone
- V.i.r. tra Carta\_premio in Evento e la chiave di Carta\_giocabile, valori nulli non ammessi

# **IMPLEMENTAZIONE DELLO SCHEMA LOGICO**

## **CREAZIONE TABELLE:**

```
CREATE TABLE Set_carta
(
Nome varchar(20) PRIMARY KEY,
Data_rilascio date
);
```

```
CREATE TABLE Gruppo
(
Nome varchar(20) PRIMARY KEY
);
```

```
CREATE TABLE Subunit
(
Nome varchar(20) PRIMARY KEY,
Gruppo_appartenenza varchar(20) NOT NULL REFERENCES Gruppo(Nome) on delete cascade
);
```

```
CREATE TABLE School_idol
(
Nome varchar(20),
Cognome varchar(20),
Data_nascita date,
Scuola varchar(20),
Anno_frequenza tinyint,
Gruppo varchar(20) NOT NULL REFERENCES Gruppo(Nome),
Subunit varchar(20) REFERENCES Subunit(Nome),
```

```
PRIMARY KEY (Nome, Cognome)
);
```

```
CREATE TABLE Carta_giocabile
(
ID char(5) PRIMARY KEY,
Rappresentazione_grafica BLOB default NULL,
Idolizzata bool NOT NULL default false,
Liv_max tinyint NOT NULL,
Skill varchar(250),
Attributo varchar(20) NOT NULL,
Rarita varchar(2) NOT NULL,
Center_skill varchar(30),
Stamina tinyint NOT NULL,
Smile int NOT NULL,
Pure int NOT NULL,
```

```
Cool int NOT NULL,  
Set_appartenenza varchar(20) REFERENCES Set_carta(Nome) on delete set NULL,  
Nome_idol varchar(20) NOT NULL,  
Cognome_idol varchar(20) NOT NULL,
```

```
FOREIGN KEY (Nome_idol, Cognome_idol) REFERENCES School_idol(Nome,  
Cognome)  
on delete cascade  
on update cascade  
);
```

```
CREATE TABLE Carta_supporto  
(  
ID char(5) PRIMARY KEY,  
Rappresentazione_grafica BLOB default NULL,  
Skill varchar(250),  
Attributo varchar(20) NOT NULL,  
Rarita varchar(2) NOT NULL,  
Set_appartenenza varchar(20) REFERENCES Set_carta(Nome) on delete set NULL,  
Nome varchar(20) NOT NULL  
);
```

```
CREATE TABLE Album  
(  
Titolo varchar(20) PRIMARY KEY,  
Data_rilascio date  
);
```

```
CREATE TABLE Canzone  
(  
Titolo varchar(20) PRIMARY KEY,  
Data_rilascio date,  
Durata varchar(20),  
Attributo varchar(20),  
Album varchar(20) REFERENCES Album(Titolo) on delete set NULL  
);
```

```
CREATE TABLE Interpretazione_solista  
(  
Nome_idol_solista varchar(20),  
Cognome_idol_solista varchar(20),  
Canzone varchar(20) REFERENCES Canzone(Titolo) on delete cascade,  
  
PRIMARY KEY (Nome_idol_solista, Cognome_idol_solista, Canzone),  
FOREIGN KEY (Nome_idol_solista, Cognome_idol_solista) REFERENCES  
School_idol(Nome, Cognome) on delete cascade  
);
```

```
CREATE TABLE Interpretazione_gruppo  
(  
Gruppo varchar(20) REFERENCES Gruppo(Nome) on delete cascade,  
Canzone varchar(20) REFERENCES Canzone(Titolo) on delete cascade,
```

```
PRIMARY KEY (Gruppo, Canzone)
);
```

```
CREATE TABLE Interpretazione_subunit
(
  Subunit varchar(20) REFERENCES Subunit(Nome) on delete cascade,
  Canzone varchar(20) REFERENCES Canzone(Titolo) on delete cascade,
```

```
PRIMARY KEY (Subunit, Canzone)
);
```

```
CREATE TABLE Evento
(
  Nome varchar(20) PRIMARY KEY,
  Tipologia varchar(20),
  Data_inizio date,
  Data_fine date,
  Stato_evento varchar(10) NOT NULL CHECK (Stato_evento in ('Passato', 'Presente',
'Futuro')),
  Carta_premio char(5) NOT NULL REFERENCES Carta_giocabile(ID),
  Punteggio_vincita int
);
```

```
CREATE TABLE Canzoni_evento
(
  Evento varchar(20) REFERENCES Evento(Nome) on delete cascade,
  Canzone varchar(20) REFERENCES Canzone(Titolo) on delete cascade,
```

```
PRIMARY KEY (Evento, Canzone)
);
```

## **RIEMPIMENTO TABELLE:**

```
INSERT INTO Set_carta(Nome, Data_rilascio) VALUES
('Seven Lucky Gods', 20151215),
('Valentines Set', 20160114),
('White Day Set', 20160215),
('Job Set part2', 20160317),
('Cyber Set', 20160414),
('Marine Set', 20160610),
('Pool Set', 20160703),
('Parents', 20160724),
('Animal ver2', 20160801),
('Victorian Set', 20160825),
('Christmas Set', 20131201),
('Bokura wa', 20150415),
('Snow Halation', 20150615),
('Wonderful Rush', 20170131),
('Rainy Season ver', 20160619),
('Initial', 20130512),
('Job set part1', 20130715),
('Animal ver1', 20130814),
('August Set', 20131010),
('New Year Set', 20140101),
('Fruit Set', 20141121),
('Maid Set', 20151010),
('Beach', 20140626);
```

```
INSERT INTO Gruppo (Nome) VALUES
('Muse'),
('Aqours'),
('Arise'),
('Wild Cats'),
('Happy Children');
```

```
INSERT INTO Subunit (Nome, Gruppo_appartenenza) VALUES
('Lily White', 'Muse'),
('Printemps', 'Muse'),
('BiBi', 'Muse'),
('Azalea', 'Aqours'),
('CYaRon', 'Aqours'),
('Guilty Kiss', 'Aqours'),
('Supersonicat', 'Wild Cats');
```

```
INSERT INTO School_idol (Nome, Cognome, Data_nascita, Scuola, Anno_frequenza,
Gruppo, Subunit) VALUES
('Rin', 'Hoshizora', 19991101, 'Otonokizaka High', 1, 'Muse', 'Lily White'),
('Hanayo', 'Koizumi', 19990117, 'Otonokizaka High', 1, 'Muse', 'Printemps'),
('Umi', 'Sonoda', 19980315, 'Otonokizaka High', 2, 'Muse', 'Lily White'),
('Maki', 'Nishikino', 19990419, 'Otonokizaka High', 1, 'Muse', 'BiBi'),
('Eli', 'Ayase', 19971021, 'Otonokizaka High', 3, 'Muse', 'BiBi'),
('Honoka', 'Kosaka', 19980803, 'Otonokizaka High', 2, 'Muse', 'Printemps'),
```



('Kotori', 'Minami', 19980912, 'Otonokizaka High', 2, 'Muse', 'Printemps'),  
 ('Nozomi', 'Tojo', 19970609, 'Otonokizaka High', 3, 'Muse', 'Lily White'),  
 ('Nico', 'Yazawa', 19970702, 'Otonokizaka High', 3, 'Muse', 'BiBi'),  
 ('Ruby', 'Kurosawa', 19990921, 'Uranoakaoshi High', 1, 'Aqours', 'CYaRon'),  
 ('Yoshiko', 'Tsushima', 19980713, 'Uranoakaoshi High', 2, 'Aqours', 'Guilty Kiss'),  
 ('Hanamaru', 'Kunikida', 19990303, 'Uranoakaoshi High', 1, 'Aqours', 'Azalea'),  
 ('Ryou', 'Ayakawa', 19970506, 'Touhou Academy', 3, 'Wild Cats', NULL),  
 ('Yurii', 'Midou', 19981225, 'Shinomome', 2, 'Happy Children', NULL),  
 ('Christina', 'Sista', 19990912, 'Touhou Academy', 3, 'Wild Cats', 'Supersonicat'),  
 ('Shun', 'Kurosaki', 19981125, 'Touhou Academy', 1, 'Wild Cats', 'Supersonicat'),  
 ('Kira', 'Tsubasa', 19970721, 'UTX', 3, 'Arise', NULL),  
 ('Elena', 'Toudou', 19970801, 'UTX', 3, 'Arise', NULL),  
 ('Yuki', 'Anjou', 19971121, 'UTX', 3, 'Arise', NULL),  
 ('Konoe', 'Kanata', 19991216, 'Shinomome', 1, 'Happy Children', NULL);

INSERT INTO Carta\_giocabile (ID, Idolizzata, Liv\_max, Skill, Attributo, Rarita,  
 Center\_skill, Stamina, Smile, Pure, Cool, Set\_appartenenza, Nome\_idol, Cognome\_idol)  
 VALUES

('0031b', 0, 40, 'Ogni 10 secondi dà la possibilità del 36% di aumentare il tuo punteggio di 200', 'Smile', 'R', 'Smile Power', 3, 2580, 1280, 1600, 'Beach', 'Umi', 'Sonoda'),  
 ('0031i', 1, 60, 'Ogni 8 secondi dà la possibilità del 36% di aumentare il tuo punteggio di 250', 'Smile', 'R', 'Smile Power', 4, 3350, 2050, 2370, 'Beach', 'Umi', 'Sonoda'),  
 ('0043b', 0, 40, 'C è la possibilità del 36% che la tua Stamina aumenti di 1 quando arrivi ad una combo di 17 note', 'Pure', 'R', 'Pure Power', 3, 1020, 3100, 1280, 'Maid Set', 'Nozomi', 'Tojo'),  
 ('0043i', 1, 60, 'C è la possibilità del 36% che la tua Stamina aumenti di 1 quando arrivi ad una combo di 15 note', 'Pure', 'R', 'Pure Power', 4, 1790, 3870, 2050, 'Maid Set', 'Nozomi', 'Tojo'),  
 ('0049b', 0, 40, 'Possibilità del 36% di aumentare il punteggio di 200 dopo 15 Perfect', 'Cool', 'R', 'Cool Power', 3, 1390, 1020, 2900, 'Wonderful Rush', 'Umi', 'Sonoda'),  
 ('0049i', 1, 60, 'Possibilità del 36% di aumentare il punteggio di 249 dopo 15 Perfect', 'Cool', 'R', 'Cool Power', 4, 2160, 1790, 3670, 'Wonderful Rush', 'Umi', 'Sonoda'),  
 ('0268b', 0, 30, NULL, 'Pure', 'N', NULL, 3, 300, 1520, 560, NULL, 'Yurii', 'Midou'),  
 ('0268i', 1, 40, NULL, 'Pure', 'N', NULL, 4, 490, 1900, 800, NULL, 'Yurii', 'Midou'),  
 ('0287b', 0, 30, NULL, 'Cool', 'N', NULL, 2, 320, 410, 1620, NULL, 'Shun', 'Kurosaki'),  
 ('0287i', 1, 40, NULL, 'Cool', 'N', NULL, 3, 390, 480, 1920, NULL, 'Shun', 'Kurosaki'),  
 ('0284b', 0, 40, 'Ogni 20 note c è la possibilità del 36% che il timing rallenti', 'Smile', 'R', 'Smile Power', 3, 3240, 1290, 1020, 'Bokura wa', 'Honoka', 'Kosaka'),  
 ('0284i', 1, 60, 'Ogni 15 note c è la possibilità del 36% che il timing rallenti', 'Smile', 'R', 'Smile Power', 4, 4010, 2060, 1790, 'Bokura wa', 'Honoka', 'Kosaka'),  
 ('0286b', 0, 40, 'Ogni 20 note c è il 36% di probabilità che il tuo punteggio aumenti di 220', 'Pure', 'R', 'Pure Power', 3, 1040, 3250, 1260, 'Bokura wa', 'Kotori', 'Minami'),  
 ('0286i', 1, 60, 'Ogni 20 note c è il 36% di probabilità che il tuo punteggio aumenti di 250', 'Pure', 'R', 'Pure Power', 4, 1810, 4020, 2030, 'Bokura wa', 'Kotori', 'Minami'),  
 ('0441b', 0, 60, 'Ogni 11 secondi c è la possibilità di recuperare 4 punti stamina', 'Cool', 'SR', 'Cool Heart', 3, 2620, 2470, 3800, 'Christmas Set', 'Rin', 'Hoshizora'),  
 ('0441i', 1, 80, 'Ogni 10 secondi c è la possibilità di recuperare 4 punti stamina', 'Cool', 'SR', 'Cool Heart', 4, 3730, 3580, 4910, 'Christmas Set', 'Rin', 'Hoshizora'),  
 ('0753b', 0, 30, NULL, 'Smile', 'N', NULL, 3, 1560, 450, 320, 'White Day Set', 'Konoe', 'Kanata'),

('0753i', 1, 40, NULL, 'Smile', 'N', NULL, 4, 1950, 560, 450, 'White Day Set', 'Konoe',  
 'Kanata'),  
 ('0802b', 0, 40, 'Ogni 10 secondi dà la possibilità del 36% di aumentare il tuo punteggio di  
 200', 'Pure', 'R', 'Pure Power', 3, 1400, 2340, 1630, 'Rainy Season ver', 'Ruby', 'Kurosawa'),  
 ('0802i', 1, 60, 'Ogni 10 secondi dà la possibilità del 36% di aumentare il tuo punteggio di  
 250', 'Pure', 'R', 'Pure Power', 4, 1560, 4030, 1830, 'Rainy Season ver', 'Ruby', 'Kurosawa'),  
 ('0803b', 0, 60, 'Ogni 15 note c è la possibilità di aumentare il tuo punteggio di 400', 'Cool',  
 'SR', 'Cool Heart', 3, 2500, 2750, 3600, 'Rainy Season ver', 'Hanamaru', 'Kunikida'),  
 ('0803i', 1, 80, 'Ogni 15 note c è la possibilità di aumentare il tuo punteggio di 450', 'Cool',  
 'SR', 'Cool Heart', 4, 2810, 3100, 4850, 'Rainy Season ver', 'Hanamaru', 'Kunikida'),  
 ('0804b', 0, 60, 'Ogni 10 note c è la possibilità del 36% che il timing rallenti', 'Smile', 'SR',  
 'Smile Power', 3, 3120, 1450, 2020, 'Rainy Season ver', 'Yoshiko', 'Tsushima'),  
 ('0804i', 1, 80, 'Ogni 8 note c è la possibilità del 36% che il timing rallenti', 'Smile', 'SR',  
 'Smile Power', 4, 3920, 1950, 2420, 'Rainy Season ver', 'Yoshiko', 'Tsushima'),  
 ('0349b', 0, 80, 'Ogni 22 perfetti c è il 35% di possibilità che il punteggio aumenti di 590',  
 'Smile', 'UR', 'Smile Princess', 4, 5000, 4305, 3620, 'Seven Lucky Gods', 'Kira', 'Tsubasa'),  
 ('0349i', 1, 100, 'Ogni 20 perfetti c è il 35% di possibilità che il punteggio aumenti di 600',  
 'Smile', 'UR', 'Smile Princess', 4, 5420, 4630, 3910, 'Seven Lucky Gods', 'Kira', 'Tsubasa'),  
 ('0350b', 0, 80, 'Ogni 17 perfect il timing rallenta per 3 secondi', 'Cool', 'UR', 'Cool  
 Princess', 4, 3640, 4230, 5010, 'Victorian Set', 'Elena', 'Toudou'),  
 ('0350i', 1, 100, 'Ogni 15 perfect il timing rallenta per 3 secondi', 'Cool', 'UR', 'Cool  
 Princess', 4, 3900, 4750, 5520, 'Victorian Set', 'Elena', 'Toudou'),  
 ('0351b', 0, 80, 'Ogni 20 secondi c è la possibilità del 10% di recuperare tutta la stamina',  
 'Pure', 'UR', 'Pure Princess', 4, 3450, 4520, 3120, 'Valentines Set', 'Yuki', 'Anjou'),  
 ('0351i', 1, 100, 'Ogni 15 secondi c è la possibilità del 20% di recuperare tutta la stamina',  
 'Pure', 'UR', 'Pure Princess', 4, 3740, 5310, 3850, 'Valentines Set', 'Yuki', 'Anjou');

INSERT INTO Carta\_supporto( ID, Rappresentazione\_grafica, Skill, Attributo, Rarita,  
 Set\_appartenenza, Nome) VALUES  
 ('0070', NULL, 'Dà la possibilità di rafforzare di poco la skill di una carta Smile', 'Smile',  
 'R', 'Animal ver2', 'Alpaca'),  
 ('0156', NULL, 'Dà la possibilità di rafforzare di molto la skill di una carta Pure', 'Pure',  
 'SR', NULL, 'Yazawa Cotaro'),  
 ('0157', NULL, 'Dà la possibilità di rafforzare di molto la skill di una carta Cool', 'Cool',  
 'SR', 'Parents', 'Nicos Mom'),  
 ('0620', NULL, 'Dà la possibilità di rafforzare di poco la skill di una carta Smile', 'Smile',  
 'R', 'Parents', 'Honokas Father');

INSERT INTO Album(Titolo, Data\_rilascio) VALUES  
 ('Snow Halation', 20131101),  
 ('Egao de Jump', 20140602),  
 ('BiBi Hits', 20160909),  
 ('Private Wars', 20130914);

INSERT INTO Canzone(Titolo, Data\_rilascio, Durata, Attributo, Album) VALUES  
 ('Private Wars', 20130914, '3:12 minuti', 'Cool', 'Private Wars'),  
 ('Start Dash', 20131112, '4:00 minuti', 'Smile', NULL),  
 ('Dance! Dance!', 20140602, '2:50 minuti', 'Smile', 'Egao de Jump'),  
 ('My demons', 20151109, '1:49 minuti', 'Cool', NULL),  
 ('Cutie Panther', 20140807, '2:58 minuti', 'Smile', 'BiBi Hits'),  
 ('Heartbeat', 20160421, '3:37 minuti', 'Cool', 'BiBi Hits'),

```

('Soldier!!', 20160909, '4:15 minuti', 'Cool', 'BiBi Hits'),
('Mystery', 20131101, '5:10 minuti', 'Pure', 'Snow Halation'),
('Love Novels', 20150603, '3:05 minuti', 'Pure', NULL),
('Loneliest Baby', 20131017, '1:57 minuti', 'Cool', 'Snow Halation'),
('Rin Rin Rin', 20140902, '1:03 minuti', 'Smile', NULL),
('Angelic Angel', 20130202, '2:05 minuti', 'Pure', NULL);

```

```

INSERT INTO Interpretazione_solista(Nome_idol_solista, Cognome_idol_solista,
Canzone) VALUES
('Rin', 'Hoshizora', 'Rin Rin Rin'),
('Rin', 'Hoshizora', 'Mystery'),
('Ruby', 'Kurosawa', 'Dance! Dance!'),
('Yuki', 'Anjou', 'Loneliest Baby'),
('Christina', 'Sista', 'Gregorian'),
('Eli', 'Ayase', 'Love Crime'),
('Hanayo', 'Koizumi', 'Love Rice'),
('Kotori', 'Minami', 'Happy'),
('Umi', 'Sonoda', 'Ocean'),
('Nico', 'Yazawa', 'iSuck'),
('Maki', 'Nishikino', 'Red'),
('Honoka', 'Kosaka', 'Bread'),
('Nozomi', 'Tojo', 'Wish'),
('Nozomi', 'Tojo', 'Prestige'),
('Yoshiko', 'Tsushima', 'Yohane'),
('Elena', 'Toudou', 'TheBest'),
('Konoe', 'Kanata', 'UfoBaby');

```

```

INSERT INTO Interpretazione_gruppo(Gruppo, Canzone) VALUES
('Muse', 'Start Dash'),
('Wild Cats', 'My Demons'),
('Arise', 'Private Wars'),
('Muse', 'Love Novels');

```

```

INSERT INTO Interpretazione_subunit(Subunit, Canzone) VALUES
('BiBi', 'Cutie Panther'),
('BiBi', 'Heartbeat'),
('BiBi', 'Soldier!!'),
('Supersonicat', 'Angelic Angel');

```

```

INSERT INTO Evento(Nome, Tipologia, Data_inizio, Data_fine, Carta_premio,
Punteggio_vincita) VALUES
('Kunikuni Summer', 'Score Match', 20160619, 20160629, '0803i', 25000),
('I wanna see you', 'Medley', 20131201, 20131212, '0441i', 22000),
('Rainbrella', 'Challenge Festival', 20160705, 20160716, '0804i', 24500),
('Bachelor Match', 'Token', 20160909, 20160920, '0351b', 11000),
('LL Sunshine', 'Medley', 20160810, 20160818, '0031b', 23000);

```

```

INSERT INTO Canzoni_evento(Evento, Canzone) VALUES
('Kunikuni Summer', 'Start Dash'),
('Kunikuni Summer', 'My Demons'),

```

('Kunikuni Summer', 'Love Novels'),  
('I wanna see you', 'My Demons'),  
('I wanna see you', 'Private Wars'),  
('I wanna see you', 'Angelic Angel'),  
('I wanna see you', 'Rin Rin Rin'),  
('I wanna see you', 'Mystery'),  
('Rainbrella', 'Cutie Panther'),  
('Rainbrella', 'Soldier!!'),  
('Rainbrella', 'Rin Rin Rin'),  
('Bachelor Match', 'Mystery'),  
('Bachelor Match', 'Heartbeat'),  
('Bachelor Match', 'Dance! Dance!'),  
('Bachelor Match', 'My Demons'),  
('LL Sunshine', 'UfoBaby'),  
('LL Sunshine', 'Prestige');

## CREAZIONE QUERY & PROCEDURE:

**QUERY1)** Trovare le canzoni che sono cantate da una subunit del gruppo 'Muse' nell'anno corrente ( 2016 ), i rispettivi album (se li hanno) e gli eventi nei quali queste sono presenti.

```
CREATE VIEW Query1 (Canzone, Album, Evento) AS
SELECT c.Titolo, a.Titolo, ce.Evento
FROM Album a RIGHT JOIN Canzone c ON a.Titolo = c.Album
JOIN Canzoni_evento ce ON c.Titolo = ce.Canzone
JOIN Interpretazione_subunit i ON c.Titolo=i.Canzone
JOIN Subunit s on i.Subunit=s.Nome
WHERE s.Gruppo_appartenenza='Muse' AND YEAR(c.Data_rilascio) = 2016;
```

### **OUTPUT:**

Canzone	Album	Evento
Heartbeat	BiBi Hits	Bachelor Match
Soldier!!	BiBi Hits	Rainbrella

**QUERY2)** Trovare Nome, Cognome e Gruppo delle idol che non hanno mai cantato nessuna canzone come solista, e di quelle che hanno almeno una loro carta rilasciata certamente prima di Natale 2014 (non comprendendo quelle con data di rilascio non nota), e delle quali almeno due canzoni (come solista) hanno partecipato ad un evento Medley.

Creazione di una vista ausiliaria che trova le School idol le cui canzoni da solista hanno partecipato almeno ad un Medley, ed inoltre il numero di tali canzoni:

```
CREATE VIEW Idol_Song_Medley(Nome_idol,Cognome_idol, Numero_canz) AS
SELECT School_idol.Nome, School_idol.Cognome, count(*)
FROM School_idol, Interpretazione_solista, Canzoni_evento, Evento
WHERE School_idol.Nome = Interpretazione_solista.Nome_idol_solista
AND School_idol.Cognome = Interpretazione_solista.Cognome_idol_solista
AND Interpretazione_solista.Canzone = Canzoni_evento.Canzone
AND Canzoni_evento.Evento = Evento.Nome
AND Evento.Tipologia = "Medley"
GROUP BY School_idol.Nome, School_idol.Cognome;
```

Creazione della query effettiva:

```
CREATE VIEW Query2 (Nome_idol, Cognome_idol, Gruppo_di_appartenenza) as
SELECT School_idol.Nome, School_idol.Cognome,Gruppo.Nome
FROM School_idol JOIN Gruppo on School_idol.Gruppo = Gruppo.Nome
WHERE School_idol.Nome not in (SELECT Nome_idol_solista FROM
Interpretazione_solista) AND Cognome not in (SELECT Cognome_idol_solista FROM
Interpretazione_solista)
```

UNION

```
SELECT SI.Nome,SI.Cognome,G.Nome
FROM School_idol SI, Gruppo G, Carta_giocabile CG, Set_carta S
WHERE SI.Gruppo = G.Nome
AND CG.Nome_idol = SI.Nome
AND CG.Cognome_idol = SI.Cognome
AND CG.Set_appartenenza = S.Nome
AND S.Data_rilascio < 20141225
AND EXISTS (
    SELECT*
    FROM Idol_Song_Medley ISM
    WHERE ISM.Nome_idol = SI.Nome AND ISM.Cognome_idol = SI.Cognome
AND Numero_canz >= 2) ;
```

**OUTPUT:**

Nome_idol	Cognome_idol	Gruppo_di_appartenenza
Hanamaru	Kunikida	Aqours
Kira	Tsubasa	Arise
Ryou	Ayakawa	Wild Cats
Shun	Kurosaki	Wild Cats
Yurii	Midou	Happy Children
Rin	Hoshizora	Muse

**TRIGGER1)** Prima dell'inserimento di una nuova carta, controlla che l'attributo di tale carta abbia la statistica corrispondente all'attributo, maggiore delle altre due.

DELIMITER ||

```
CREATE TRIGGER Controllo_attributi
BEFORE INSERT ON Carta_giocabile
FOR EACH ROW
BEGIN
    if(NEW.Attributo = 'Smile') THEN
        if(new.Smile < new.Cool OR new.Smile < new.Pure )THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT ='SMILE minore di PURE o COOL';
        END IF;
    END IF;

    if(NEW.Attributo = 'Cool') THEN
        if(new.Cool < new.Smile OR new.Cool < new.Pure )THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT ='COOL minore di PURE o SMILE';
        END IF;
    END IF;

    if(NEW.Attributo = 'Pure') THEN
        if(new.Pure < new.Smile OR new.Pure < new.Cool )THEN
```

```

        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'PURE minore di COOL o SMILE';
    END IF;
END IF;
END ||

```

**TRIGGER2)** Prima dell'inserimento di un nuovo evento, questo trigger determina in automatico lo stato degli eventi quando questi vengono inseriti, e lo inserisce di conseguenza.

```

DELIMITER ||

```

```

CREATE TRIGGER Riempi_statoevento
BEFORE INSERT ON Evento
FOR EACH ROW
BEGIN
    IF(new.Data_fine < CURDATE()) THEN
        SET new.Stato_evento = 'Passato';
    END IF;

    IF(new.Data_inizio <= CURDATE() AND new.Data_fine >= CURDATE()) THEN
        SET new.Stato_evento = 'Presente';
    END IF;

    IF(new.Data_inizio > CURDATE()) THEN
        SET new.Stato_evento = 'Futuro';
    END IF;
END ||

```

**FUNZIONE1)** Funzione che restituisce, per una idol specifica, la media di uno dei suoi tre statistiche, prendendo in considerazione tutte le carte con tale idol.

```

DELIMITER ||

```

```

CREATE FUNCTION Media_attributi(NomeIdol varchar(20),CognomeIdol
varchar(20),Stat varchar(20))
RETURNS decimal(6,2)
BEGIN
    DECLARE media decimal(6,2);

    IF(Stat ='Smile') THEN
        SELECT avg(Smile) into media
        FROM Carta_giocabile
        WHERE Nome_idol = NomeIdol && Cognome_idol = CognomeIdol;
    END IF;

    IF(Stat ='Cool') THEN
        SELECT avg(Cool) into media
        FROM Carta_giocabile
        WHERE Nome_idol = NomeIdol && Cognome_idol = CognomeIdol;
    END IF;

```

```

        IF(Stat ='Pure') THEN
        SELECT avg(Pure) into media
        FROM Carta_giocabile
        WHERE Nome_idol = NomeIdol && Cognome_idol = CognomeIdol;
        END IF;

        RETURN media;
END ||

```

**FUNZIONE2)** Funzione che, data una certa statistica e una rarità, restituisce l'ID della Carta con il valore massimo di quella statistica, e avente tale rarità.

```

DELIMITER ||

```

```

CREATE FUNCTION Carta_maggiore(statistica varchar(20), valore varchar(20))
RETURNS char(5)
BEGIN
    DECLARE Best char(5);

    IF(statistica = "Smile") THEN
        SELECT ID into Best
        FROM Carta_giocabile
        WHERE Smile = (SELECT MAX(Smile) FROM School_idol) AND Rarita =
valore
        LIMIT 1;
        END IF;

    IF(statistica = "Cool") THEN
        SELECT ID into Best
        FROM Carta_giocabile
        WHERE Cool = (SELECT MAX(Cool) FROM School_idol) AND Rarita = valore
        LIMIT 1;
        END IF;

    IF(statistica = "Pure") THEN
        SELECT ID into Best
        FROM Carta_giocabile
        WHERE Pure = (SELECT MAX(Pure) FROM School_idol) AND Rarita = valore
LIMIT 1;
        END IF;

    RETURN Best;
END ||

```



**PROCEDURA)** Procedura che mi permette di aumentare o diminuire una statistica di una carta di una certa percentuale.

DELIMITER ||

CREATE PROCEDURE ModificaStat\_percentuale(idCarta char(5), attributo varchar(20), percentuale int)

BEGIN

IF(attributo = "Smile") THEN  
UPDATE Carta\_giocabile  
SET Smile = Smile + (Smile\*percentuale)/100  
WHERE ID = idCarta;  
END IF;

IF(attributo = "Cool") THEN  
UPDATE Carta\_giocabile  
SET Cool = Cool + (Cool\*percentuale)/100  
WHERE ID = idCarta;  
END IF;

IF(attributo = "Pure") THEN  
UPDATE Carta\_giocabile  
SET Pure = Pure + (Pure\*percentuale)/100  
WHERE ID = idCarta;  
END IF;

END||