Progetto Basi Di Dati

PokéSquad

Marco Nardelotto – Matricola 1189563

Marco Positello – Matricola

****

**Abstract:**

Pokémon è un media franchise giapponese di proprietà della The Pokémon Company, creato nel 1996 da Satoshi Tajiri. Esso è incentrato su delle creature immaginarie chiamate appunto "Pokémon", che gli umani possono catturare, allenare e far combattere tra di loro per scopi ludici. Il franchise nasce come coppia di videogiochi sviluppati da Game Freak e pubblicati da Nintendo per la console portatile Game Boy. L'enorme successo ottenuto in tutto il mondo ha portato alla creazione di ulteriori videogiochi, anime, film, manga, un gioco di carte collezionabili, libri e innumerevoli gadget e giocattoli.

Il progetto PokèSquad, dato l’incremento della passione da parte di molte persone al mondo dei Pokémon dovuto al rilascio di nuove applicazioni e videogames che si basano su questo franchise, ha l’obiettivo di mettere a disposizione per tutti gli appassionati di questa saga una base di dati per semplificare la ricerca e la gestione delle oltre 800 specie Pokémon esistenti, di tutte le informazioni relative ad essi e di tutto ciò che è compreso all’interno del loro mondo videoludico. Sarà inoltre utile per aiutare gli aspiranti allenatori a gestire i propri “mostriciattoli” catturati nell’arco della loro avventura, avendo anche la possibilità di suddividerli in squadre e per monitorare meglio l’andamento dell’allenamento con i proprio Pokémon.

**Sommario**

[1 Progettazione Concettuale 2](#_Toc222538)

[1.1 Analisi dei requisiti 2](#_Toc222539)

[1.1.1 Descrizione Testuale 2](#_Toc222540)

[1.1.2 Descrizione delle Entità 4](#_Toc222541)

[1.1.3 Relazioni 7](#_Toc222542)

[1.1.4 Generalizzazioni 8](#_Toc222543)

[1.1.5 Descrizione dei vincoli di integrità aggiuntivi 8](#_Toc222544)

[1.2 Schema ER 8](#_Toc222545)

[2 Progettazione Logica 9](#_Toc222546)

[2.1 Ristrutturazione Entità-Relazioni 9](#_Toc222547)

[2.1.1 Eliminazione Generalizzazioni 9](#_Toc222548)

[2.1.1.1 Generalizzazione Zona Cattura 9](#_Toc222549)

[2.1.1.2 Generalizzazione Strumenti 9](#_Toc222550)

[2.1.2 Altre modifiche allo schema 9](#_Toc222551)

[2.1.2.1 Scomposizione Tipo – Efficacia 9](#_Toc222552)

[2.1.2.2 Relazione Linea Evolutiva 9](#_Toc222553)

[2.2 Ristrutturazione Schema E/R 10](#_Toc222554)

[2.3 Traduzione verso il modello logico 11](#_Toc222555)

[2.3.1 Schema logico 11](#_Toc222556)

[2.3.2 Schema logico – relazionale 11](#_Toc222557)

[3 Implementazione 12](#_Toc222558)

[3.1 Query 12](#_Toc222559)

[3.1.1 Query N°1 12](#_Toc222560)

[3.1.2 Query N°2 12](#_Toc222561)

[3.1.3 Query N°3 13](#_Toc222562)

[3.1.4 Query N°4 13](#_Toc222563)

[3.1.5 Query N°5 13](#_Toc222564)

[3.1.6 Query N°6 14](#_Toc222565)

[3.1.7 Query N°7 14](#_Toc222566)

[3.1.8 Query N°8 14](#_Toc222567)

[3.2 Procedure 14](#_Toc222568)

[3.2.1 Procedura N°1 14](#_Toc222569)

[3.3 Funzioni 14](#_Toc222570)

[3.3.1 Funzione N°1 14](#_Toc222571)

[3.3.2 Funzione N°2 15](#_Toc222572)

[3.4 Trigger 15](#_Toc222573)

[3.4.1 Trigger N°1 a 15](#_Toc222574)

[3.4.2 Trigger N°1 b 15](#_Toc222575)

[3.4.3 Trigger N°2 a 15](#_Toc222576)

[3.4.4 Trigger N°2 b 15](#_Toc222577)

# Progettazione Concettuale

## Analisi dei requisiti

### Descrizione Testuale

Si vuole realizzare una base di dati per l’organizzazione e la gestione dei Pokémon e delle squadre di cui fanno parte.

Ciascun Pokémon fa parte di una specie delle quali è necessario conoscere:

* Un codice identificativo univoco
* Un nome
* Peso
* Altezza
* Breve descrizione
* Tipo/i
* Tasso di cattura: probabilità percentuale che ha l’allenatore di catturare un Pokémon di questa specie
* Affetto di base
* Statistiche di base formate da:
  + Punti salute
  + Attacco
  + Difesa
  + Attacco Speciale
  + Difesa Speciale
  + Velocità
* Mosse
* Evoluzioni: evoluzione di un Pokémon da una forma primitiva ad una forma più avanzata. (Un Pokémon può anche non avere evoluzioni)

Ciascuna specie può avere al massimo 2 tipi e per ogni tipo sono note le debolezze e le resistenze.

Ogni specie ha delle statistiche di base fisse che possono migliorare con l’aumentare del livello.

Ogni specie ha un elenco di mosse che può imparare

Ciascun Pokémon ha:

* Un nickname: nome che l’allenatore dà al Pokémon dopo la cattura
* Livello: livello d’esperienza relativo al Pokémon
* Strumento: oggetto assegnato al Pokémon che potrà essere utilizzato da esso
* Mosse imparate: può conoscere al massimo 4 mosse tra quelle che esso può imparare
* Affetto attuale: valore che indica l’amicizia verso l’allenatore
* Statistiche attuali formate da:
  + Punti salute
  + Attacco
  + Difesa
  + Attacco Speciale
  + Difesa Speciale
  + Velocità
* Abilità: capacità passiva del Pokémon
* Natura: meccanica di gioco che può alterare i valori di 2 statistiche, una in meglio e una in peggio.
* Zona di cattura

Un Pokémon può possedere solamente uno strumento alla volta, può avere solo una natura e una sola abilità ed infine una sola zona di cattura.

Per ciascuno strumento è necessario conoscere il nome, una breve descrizione per l’utilizzo e il tipo che può essere:

* Base
* Bacca
* Rimedi
* MT (macchina tecnica)

Le statistiche attuali tengono traccia del miglioramento delle statistiche di base.

Il livello di un Pokémon può andare da 1 a 100.

Il nickname è univoco.

Zona di cattura può essere:

* Percorso
* Grotta
* Edificio

Per ciascuna mossa è necessario conoscere:

* Nome
* Potenza (da -2 a 250): con -1 si indicano le mosse che mandano sempre K.O. il bersaglio, con -2 si indica che il danno non è fisso e può essere calcolato solo in battaglia.
* Tipo
* Categorie che possono essere:
  + Fisica
  + Speciale
  + Status
* Precisione
* PP (punti potenza) base (1-30): indicano quante volte la mossa può essere usata in battaglia
* Breve descrizione

Ciascuna specie di Pokémon fa parte di almeno una generazione della quale è necessario conoscere:

* Numero identificativo sequenziale
* Nome
* Regione

Ciascun Pokémon può far parte di una o più squadre di cui devo conoscere:

* Nome
* Componenti
* Numero di vittorie
* Numero di sconfitte

Una squadra deve avere 6 componenti distinti.

### Descrizione delle Entità

**Specie Pokemon**

Ogni pokémon appartiene a una specie basata sulle sue caratteristiche biologiche

Attributi

* ID: Varchar(10) ​**PK** ​​- I​d univoco che identifica ogni specie pokémon
* Nome: varchar(20) - ​Nome della specie pokémon
* Peso: Decimal(4,1) - ​Peso in grammi di una specie pokémon
* Altezza: Decimal(3,1) - ​Altezza in centimetri di una specie pokémon
* Tipo1 smallint **FK**(Tipo) - ​Primo tipo una specie pokémon
* Tipo2 smallint **FK**(Tipo) - ​Secondo tipo (opzionale) di una specie pokémon
* Descrizione varchar(50)
* TassoDiCattura Decimal(4,2) - valore percentuale tra 1 e 100
* Affetto di base smallint
* PuntiSaluteBase smallint – valore numerico per identificare i punti salute base
* AttaccoBase smallint – valore numerico per identificare l’attacco base
* DifesaBase smallint – valore numerico per identificare la difesa base
* AttSpBase smallint – valore numerico per identificare l’attacco speciale base
* DifSpBase smallint – valore numerico per identificare la difesa speciale base
* VelBase smallint – valore numerico per identificare la velocità base

**Mosse**

Una mossa è un attacco che il Pokémon può usare in battaglia

Attributi (si può aggiungere introduzione mossa)

* IdMossa smallint ​**PK** ​​- I​d univoco che identifica ogni mossa
* NomeMossa varchar(15)
* Potenza smallint – quantità di danno inflitto
* TipoMossa smallint **FK**(Tipo) – tipo della mossa
* Categoria varchar(15) – fisica,speciale,status
* Precisione smallint – da 1 a 100 (null indica sempre a segno)
* PPBase smallint – da 1 a 60
* DescrMossa varchar(150)

**Evoluzione**

Un Pokémon può evolversi in un altro, attraverso una modalità di evoluzione

Attributi

* IdPokemonEvoluto Varchar(10) **PK**– in cosa si evolve il pokemon
* idPokemonEvolvente Varchar(10) – da chi si evolve
* ModalitaEvoluzione varchar(50) – come avviene l’evoluzione
* LivelloEvoluzione smallint – non obbligatorio
* StatoEvoluzione tinyint – quante pre-evoluzioni ha il Pokémon

**Pokemon**

Questa entità definisce i Pokémon catturati e quindi posseduti dall’allenatore

Attributi

* Nickname varchar(20) **PK** – soprannome del pokemon posseduto
* CeSpecie Varchar(10) **FK**(SpeciePokemon) – specie di appartenenza
* Livello smallint – da 1 a 100
* CeStrumento smallint **FK**(Strumenti) – strumento tenuto dal pokemon
* AffettoAttuale smallint - >= di affetto base
* PuntiSaluteAttu smallint – >= al valore numerico di punti salute base
* AttaccoAttu smallint – >= al valore numerico di attacco base
* DifesaAttu smallint – >= al valore numerico di difesa base
* AttSpAttu smallint – >= al valore numerico di attacco speciale base
* DifSpAttu smallint – >= al valore numerico di difesa speciale base
* VelAttu smallint – >= al valore numerico di velocità base
* CeAbilita smallint **FK**(Abilita) – caratteristica del pokemon
* CeNatura smallint **FK**(Natura) – caratteristica del pokemon
* CeZonaCattura smallint **FK**(ZonaCattura)

**Zona Cattura**

Un Pokémon viene catturato in una determinata zona della regione\*

Attributi

* IdZona smallint **PK**
* NomeZona varchar(25)
* DescrZona varchar(150)
* Morfologia varchar(10) – caratteristiche fisiche della zona

**Generazione**

Definisce i cambiamenti del gioco nel corso delle varie generazioni

Attributi

* IdGen smallint **PK**
* NomeGen varchar (20)
* Regione varchar(15)

**Squadre**

I Pokémon catturati possono essere assegnati a squadre composte da 6 Pokémon ciascuna

Attributi

* NomeSquadra varchar(20) **PK**
* NumeroVittorie smallint - >=0
* NumeroSconfitte smallint - >=0

**Abilità**

Una abilità è una capacità passiva del Pokémon che viene resa nota al momento della cattura

Attributi

* IdAbilita smallint **PK**
* NomeAbilita varchar(20)
* DescrAbilita varchar(100)
* Introduzione smallint **FK**(Generazione) – definisce in che generazione è stata introdotta l’abilita

**Natura**

Una Natura è una capacità passiva del Pokémon che viene resa nota al momento della cattura

Attributi

* IdNatura smallint **PK**
* NomeNatura varchar(20)
* StatMigliorata varchar(20) – scelta tra le statistiche, può essere null se non migliora niente
* StatPeggiorata varchar(20) – scelta tra le statistiche, può essere null se non peggiora niente
* SaporeAmato varchar(20) - gusto che preferisce il pokemon con questa natura
* SaporeOdiato varchart(20) - gusto che NON preferisce il pokemon con questa natura

**Tipo**

Un tipo è una proprietà che caratterizza un pokémon o una mossa. Ogni tipo può essere efficace, debole, neutrale o immune verso altri tipi.

Attributi

* IdTipo smallint **PK**
* NomeTipo varchart(25)

**Strumenti**

Uno Strumento è un oggetto che può essere affidato a un Pokémon che poi lo utilizzerà in battaglia

Attributi

* IdStrum smallint PK
* NomeStrum varchar(25)
* DescrStrum varchar(150)
* TipoStrum varchar(20) - scelto tra Base, Bacca, Rimedio, MT (Macchina Tecnica)

### Relazioni

**Apprendimento: Specie Pokémon – Mosse**

* Lista delle mosse che può apprendere quella specie di Pokémon

**Tipologia Pokémon: Specie Pokémon – Tipo**

* Di quali tipi (1 o 2 massimo) fa parte una specie di Pokémon

**Affiliazione: Specie Pokémon – Pokémon**

* A quale Specie Pokémon fa parte un determinato Pokémon catturato

**Linea evolutiva: Specie Pokémon - Evoluzione**

* Certi Pokémon (base) possono evolversi in altri Pokémon (evoluzione)
* Un Pokémon può essere contemporaneamente sia base che evoluzione
* Un Pokémon evoluzione ha al più una base

**Strumento posseduto: Pokémon - Strumenti**

* Un Pokémon può tenere al massimo uno strumento

**Caratterizzazione Uno: Pokémon – Abilità**

* Un Pokèmon ha solo una abilità

**Caratterizzazione Due: Pokémon – Natura**

* Un Pokémon ha solo una natura

**Conoscenza: Pokémon – Mosse**

* Il Pokémon conosce fino a 4 mosse tra quelle che può imparare

**Habitat: Pokémon – Zona Cattura**

* Un Pokémon viene catturato in una specifica zona

**Introduzione: Abilità – Generazione**

* Un’ Abilità è stata introdotta in una generazione

**Tipologia Mossa: Tipo – Mossa**

* Una Mossa è di un Tipo specifico

**Efficacia: Tipo – Tipo**

* Un Tipo può avere un modificatore di efficacia nei confronti di un altro Tipo
* Di default il modificatore è 1x

**Composizione: Pokémon – Squadre**

* Un Pokémon può essere inserito in una squadra
* Ogni Squadra comprende 6 Pokémon

### Generalizzazioni

**Zona Cattura**

**Percorso – Grotta – Edificio**

Una Zona si distingue per le sue caratteristiche morfologiche (g. completa)

**Strumenti**

**Base – Bacche – Rimedi – MT**

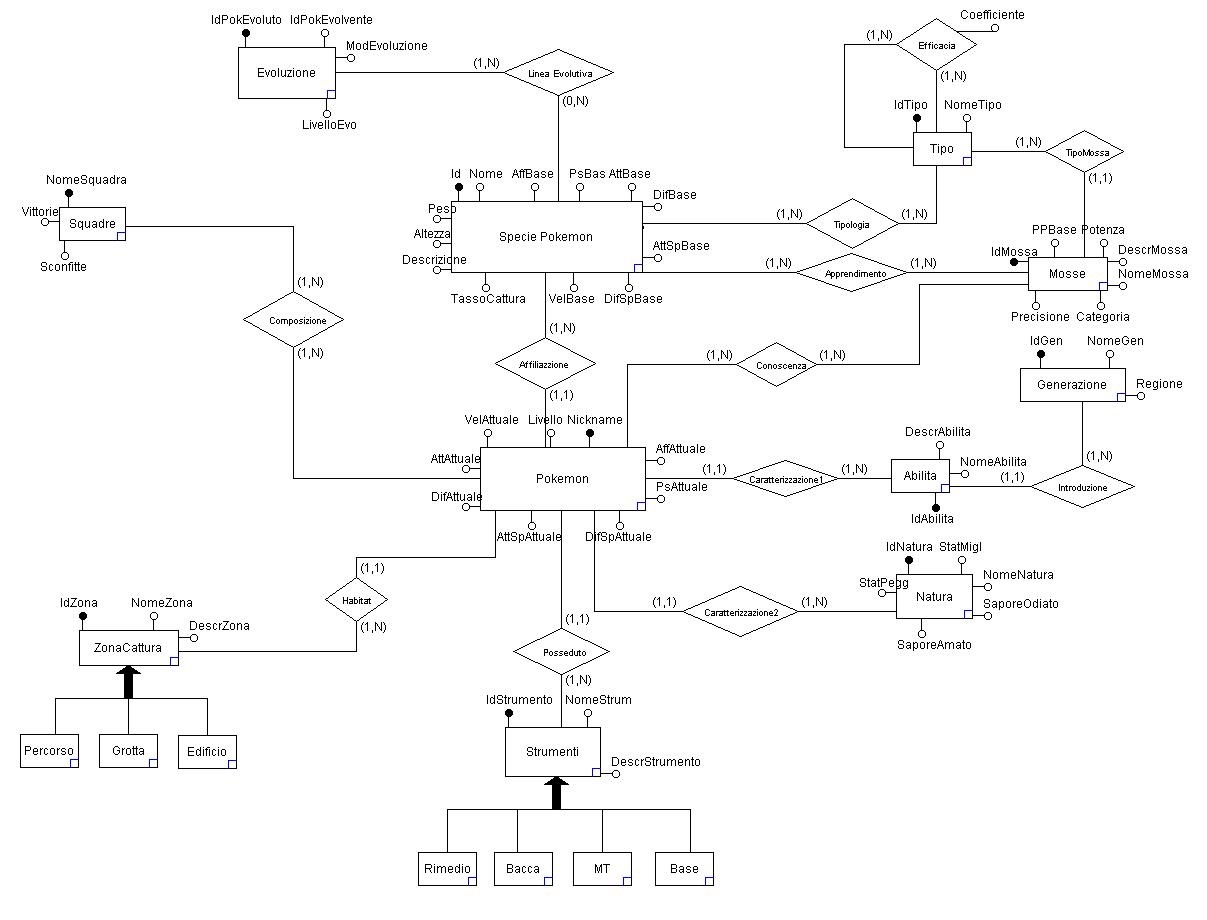
Uno strumento viene distinto dalla sua categoria di appartenenza

### Descrizione dei vincoli di integrità aggiuntivi

Lo stato dell’evoluzione è un intero compreso fra [-1,4], (-1 = pokémon baby; 0 = pokémon base, 1°,2°,3°,4° stato evolutivo (nb: non esistono ancora pokémon con stati evolutivi > =3)

Il coefficiente nella tabella efficacia può essere solamente compreso tra [0,2] ed è sempre un multiplo di 0.5.

## Schema ER



# Progettazione Logica

## Ristrutturazione Entità-Relazioni

### Eliminazione Generalizzazioni

#### Generalizzazione Zona Cattura

La generalizzazione riguardante l’entità ZonaCattura viene trasformata in un attributo dal nome “morfologia”, che ne descrive il tipo di zona, inserito nella medesima entità. Questa scelta è dovuta al fatto che una zona possiede una sola caratteristica morfologica, quindi non c’è ridondanza nei dati.

#### Generalizzazione Strumenti

Come nel caso della generalizzazione *“ZonaCattura”* le entità Rimedio, Base, MT, Bacca vengono eliminate e viene inserito un attributo (tipoStrumento) all’interno dell’entità Strumenti, tale attributo potrà contenere solo uno dei precedenti quattro valori. Questa scelta, come quella precedente, è dovuta al fatto che uno zona strumento appartiene solo a una delle quattro categorie descritte, quindi non c’è ridondanza nei dati.

### Altre modifiche allo schema

#### Scomposizione Tipo – Efficacia

Per gestire l’efficacia di ogni mossa è necessario partizionare l’entità tipo in 2 Entità:

* Tipo: avente gli stessi attributi di prima (IdTipo, NomeTipo)
* Efficacia: avente l’attributo coefficiente (di tipo Decimal(1,1) e 2 chiavi primarie ed esterne:
  + - Attaccante: determina il tipo attaccante
    - Difensore: determina il tipo difensore

Ci saranno poi 2 relazioni che associano le 2 attività: attacco e difesa.

Siccome i tipi Pokémon non sono soggetti ad aggiornamenti frequenti, in caso di aggiunta di un nuovo tipo, sarà necessario disabilitare i controlli sulle chiavi esterne e aggiungere le voci mancanti alle varie tabelle manualmente.

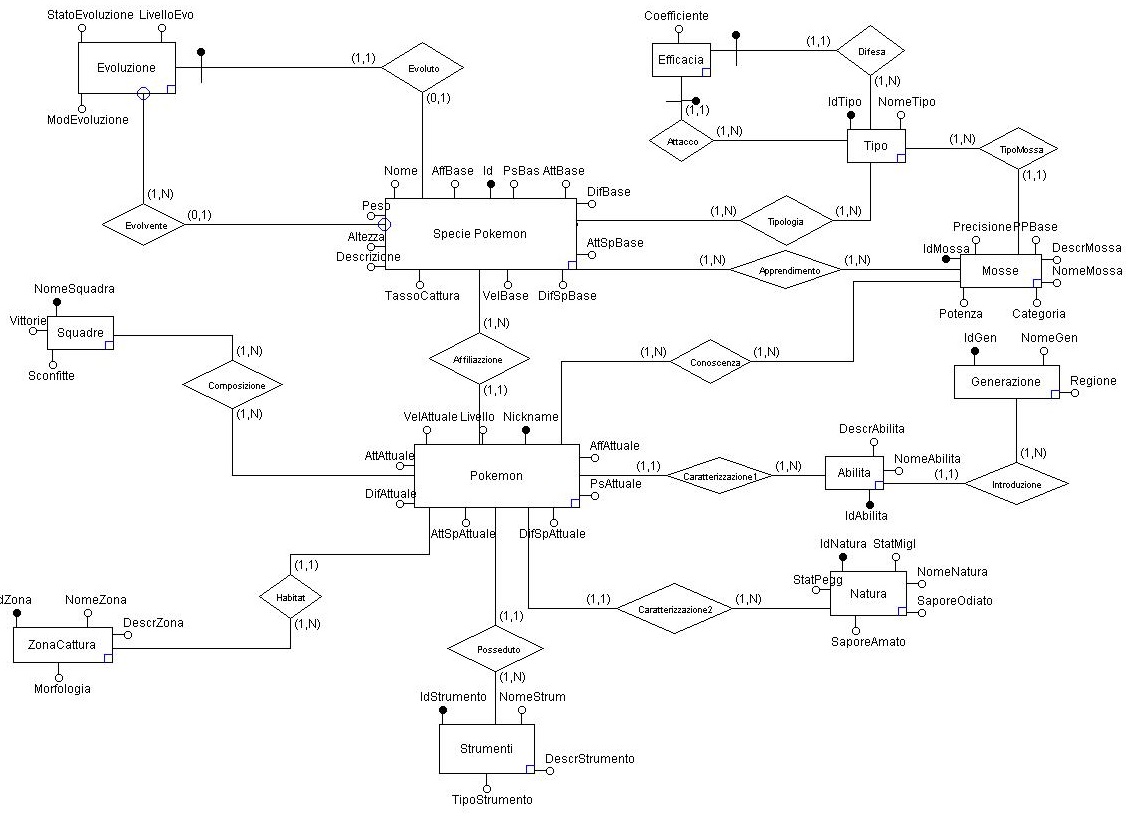
#### Relazione Linea Evolutiva

La relazione “Linea Evolutiva” è stata divisa in 2 relazioni:

* Evoluto: specifica l’evoluzione finale del Pokémon evolvente
* Evolvente: specifica il Pokémon da cui inizia l’evoluzione

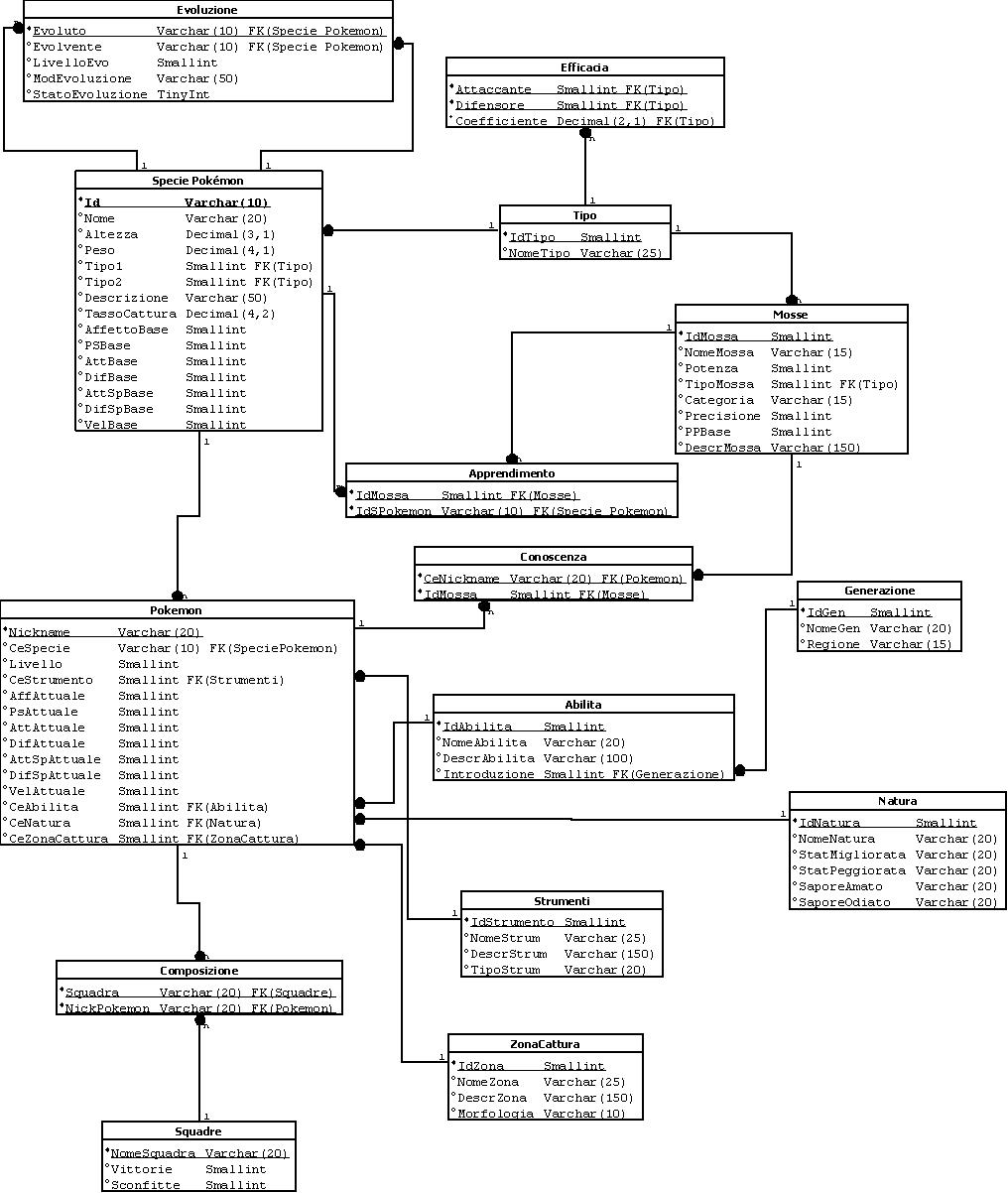
Le due entità delle relazioni rimangono inalterate con la differenza che la chiave primaria di Evoluzione sarà anche chiave esterna della relazione Evoluto con Specie Pokémon, inoltre è stato aggiunto l’attributo StatoEvoluzione per identificare a che stage sia arrivata l’evoluzione.

## Ristrutturazione Schema E/R



## Traduzione verso il modello logico

### Schema logico



### Schema logico – relazionale

* Attributi Sottolineati indicano delle chiavi
* Attributi in Corsivo indicano chiavi esterne

**SpeciePokemon** (Id, Nome, Altezza, Peso, *Tipo1*, *Tipo2* (0,1), Descrizione, TassoCattura, AffettoBase, PSBase, AttBase, DifBase, AttSpBase, DifSpBase, VelBase)

**Pokemon** (Nickname, *CeSpecie*, Livello, *CeStrumento*, AffAttuale, PsAttuale, AttAttuale, DifAttuale, AttSpAttuale, DifSpAttuale, VelAttuale, *CeAbilita*, *CeNatura*, *CeZonaCattura*)

**Evoluzione** (*Evoluto*, *Evolvente*, LivelloEvo, ModEvoluzione, StatoEvoluzione)

**Tipo** (IdTipo, NomeTipo)

**Efficacia** (*Attaccante*, *Difensore*, Coefficiente)

**Mosse** (IdMossa, NomeMossa, Potenza, *TipoMossa*, Categoria, Precisione, PPBase, DescrMossa)

**Apprendimento** (*IdMossa*, *IdSPokemon*)

**Conoscenza** (*CeNickname*, *IdMossa*)

**Generazione** (IdGen, NomeGen, Regione)

**Abilita** (IdAbilita, NomeStrum, DescrAbilita, *Introduzione*)

**Natura** (IdNatura, NomeNatura, StatMigliorata, StatPeggiorata, SaporeAmato, SaporeOdiato)

**Strumenti** (IdStrumento, NomeStrum, DescrStrum, TipoStrum)

**ZonaCattura** (IdZona, NomeZona, DescrZona, Morfologia)

**Composizione** (*Squadra*, *NickPokemon*)

**Squadre** (NomeSquadra, Vittorie, Sconfitte)

# Implementazione

## Query

### Query N°1

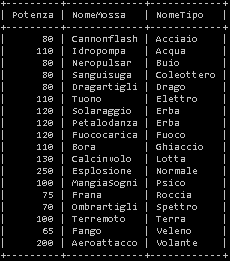
Query che rappresenta la mossa con la potenza maggiore per ogni tipo esistente

**SELECT** Potenza**,**NomeMossa**,**NomeTipo

**FROM** mosse **as** x **INNER** **JOIN** tipo **ON** x**.**TipoMossa**=**tipo**.**IdTipo

**WHERE** Potenza **=** **(SELECT** **MAX(**Potenza**)** **FROM** mosse **as** y

**GROUP** **BY** y**.**TipoMossa **HAVING** x**.**TipoMossa**=**y**.**TipoMossa**);**



### Query N°2

Query che visualizza i pokemon che compongono la squadra con il maggior numero di vittorie

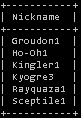
**SELECT** Nickname

**FROM** pokemon **INNER** **JOIN** composizione **ON** pokemon**.**Nickname**=**composizione**.**NickPokemon **INNER** **JOIN** squadre **ON** composizione**.**Squadra**=**squadre**.**NomeSquadra

**WHERE** NomeSquadra**=(SELECT** NomeSquadra

**FROM** squadre

**WHERE** vittorie **=** **(SELECT** **MAX(**vittorie**)** **FROM** squadre**));**

****

### Query N°3

Query che restituisce i nomi delle specie pokémon che apprendono le mosse con il più alto tasso di apprendimento

**SELECT** **DISTINCT(**id**),**nome

**FROM** speciepokemon **INNER** **JOIN** apprendimento **on** speciepokemon**.**Id**=**apprendimento**.**IdSPokemon

**WHERE** IdMossa **=** **any(SELECT** IdMossa

**FROM** apprendimento

**GROUP** **BY** IdMossa

**HAVING** **COUNT(\*)=(SELECT** **Max(**x**.**tot**)**

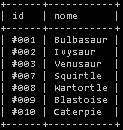
**FROM** **(SELECT** **COUNT(\*)** **as** tot

**FROM** apprendimento

**GROUP** **BY** IdMossa

**)** x**))**

**ORDER** **BY** id**;**



### Query N°4

Query che restituisce i nomi delle specie pokémon e il numero delle mosse del loro stesso tipo principale (tipo1) che possono apprendere

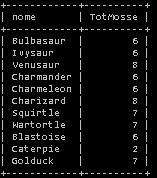
**SELECT** nome**,** **COUNT(**nomeMossa**)** **AS** TotMosse

**FROM** speciePokemon **INNER** **JOIN** tipo **as** T1 **ON** speciepokemon**.**Tipo1**=**T1**.**IdTipo **INNER** **JOIN** apprendimento **ON** speciepokemon**.**Id**=**apprendimento**.**IdSPokemon **INNER** **JOIN** mosse **ON** apprendimento**.**IdMossa**=**mosse**.**IdMossa **INNER** **JOIN** tipo **AS** T2 **ON** mosse**.**TipoMossa**=**T2**.**IdTipo

**WHERE** speciepokemon**.**Tipo1**=**mosse**.**TipoMossa

**GROUP** **BY** nome

**ORDER** **BY** id**;**



### Query N°5

Query che restituisce le specie pokémon che si evolvono tramite l’uso di pietre

**SELECT** S**.**Nome **as** PokemonEvoluto**,** S2**.**Nome **as** EvolveDa**,** E**.**ModEvoluzione **as** Tramite

**FROM** Evoluzione **as** E **JOIN** SpeciePokemon **as** S **JOIN** SpeciePokemon **as** S2

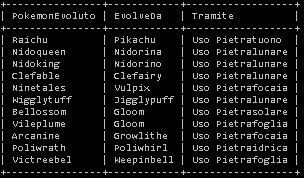
**ON** E**.**Evoluto **=** S**.**id

**WHERE** E**.**ModEvoluzione **like** "Uso Pietra%"

**AND** E**.**Evoluto **=** S**.**id

**AND** E**.**Evolvente **=** S2**.**id

**ORDER** **BY** S2**.**id**;**

****

### Query N°6

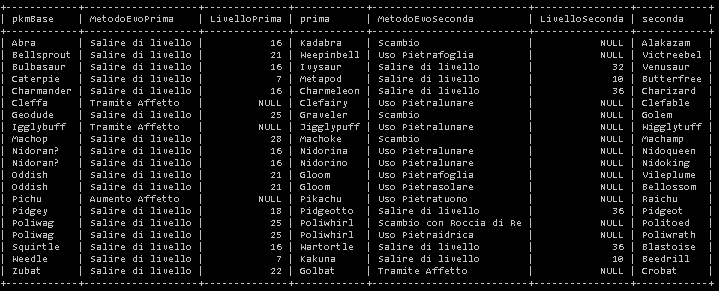
Query che restituisce i dati dei pokemon che hanno 3 evoluzioni e i relativi metodi di evoluzione

**SELECT** pokeName**(**base**.**Evolvente**)** **AS** pkmBase**,** base**.**ModEvoluzione **as** MetodoEvoPrima**,** base**.**LivelloEvo **as** LivelloPrima**,** pokeName**(**base**.**Evoluto**)** **AS** prima**,**ev**.**ModEvoluzione **as** MetodoEvoSeconda**,** ev**.**LivelloEvo **as** LivelloSeconda**,** pokeName**(**ev**.**Evoluto**)** **AS** seconda

**FROM** evoluzione **as** base **INNER** **JOIN** evoluzione **as** ev **ON** base**.**Evoluto**=**ev**.**Evolvente

**WHERE** ev**.**StatoEvoluzione **=** 2

**ORDER** **BY** pkmBase;



## Procedure

### Procedura N°1

Procedura che dato l’id di un tipo restitusce le mosse che sono molto efficaci contro quel tipo

**CREATE** **PROCEDURE** TipoDebMosse **(type** **SMALLINT)**

**BEGIN**

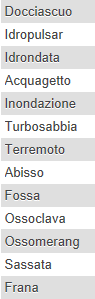
**SELECT** nomeMossa **FROM** mosse **as** m

**WHERE** m**.**TipoMossa **=** **Any** **(SELECT** e**.**Attaccante

**FROM** tipo **INNER** **JOIN** efficacia **as** e **ON** tipo**.**IdTipo**=**e**.**Difensore

**WHERE** e**.**Difensore**=type** **AND** e**.**Coefficiente**>**1.0**);**

**END**

**** Il comando eseguito ponendo come paramentro type=1

## Funzioni

### Funzione N°1

Funzione che dato un tipo restituisce il numero di mosse di quel tipo

**CREATE** **FUNCTION** MosseTipo**(**tipoM **varchar(**25**))**

**RETURNS** **tinyint**

**BEGIN**

**DECLARE** NumMosseTipo **tinyint;**

**SELECT** **count(\*)** **into** NumMosseTipo

**FROM** mosse **as** m **JOIN** apprendimento **as** a **on** m**.**IdMossa **=** a**.**IdMossa **JOIN** tipo **AS** t **ON** t**.**IdTipo **=** m**.**TipoMossa

**WHERE** t**.**NomeTipo **=** TipoM**;**

**RETURN** NumMosseTipo**;**

**END**

### Funzione N°2

Funzione che data una natura restituisce il numero di pokemon di quella natura

**CREATE** **FUNCTION** NaturePok**(**NaturaP **varchar(**25**))**

**RETURNS** **tinyint**

**BEGIN**

**DECLARE** NumPk **tinyint;**

**SELECT** **count(\*)** **into** NumPk

**FROM** natura **as** n **JOIN** pokemon **as** p **on** n**.**IdNatura **=** p**.**CeNatura

**WHERE** n**.**IdNatura **=** NaturaP**;**

**RETURN** NumPk**;**

**END**

### Funzione N°3

Funzione che resituisce il nome di un pokemon dato l’id

**CREATE** **FUNCTION** pokeName**(**pkId **varchar(**15**))**

**BEGIN**

**RETURN** **(SELECT** Nome **FROM** SpeciePokemon **AS** s **WHERE** s**.**ID **=** pkID**);**

**END**

## Trigger

### Trigger N°1 a

Verifica che un livello da inserire non sia maggiore di 100 in tal caso lo setta a 50.

**CREATE** **TRIGGER** InsertValidaLivello **BEFORE** **INSERT** **on** pokemon

**FOR** **EACH** **ROW**

**BEGIN**

**if(new.**Livello **>** 100**)** **THEN** **SET** **new.**Livello **=** 50**;**

**end** **if;**

**END**

### Trigger N°1 b

Verifica che un livello da modificare non sia maggiore di 100 in tal caso lo setta a 50.

**CREATE** **TRIGGER** UpdateValidaLivello **BEFORE** **UPDATE** **on** pokemon

**FOR** **EACH** **ROW**

**BEGIN**

**if(new.**Livello **>** 100**)** **THEN** **SET** **new.**Livello **=** 50**;**

**end** **if;**

**END**

### Trigger N°2 a

Verifica che una potenza da inserire non sia maggiore di 250 o minore di -2 in tal caso la setta a -2.

**CREATE** **TRIGGER** InsertValidaPotenza **BEFORE** **INSERT** **on** mosse

**FOR** **EACH** **ROW**

**BEGIN**

**if(new.**potenza **>** 250 **OR** **new.**potenza **<** -2**)** **THEN** **SET** **new.**Potenza **=** -2**;**

**end** **if;**

**END**

### Trigger N°2 b

Verifica che una potenza da modificare non sia maggiore di 250 o minore di -2 in tal caso la setta a -2.

**CREATE** **TRIGGER** UpdateValidaPotenza **BEFORE** **UPDATE** **on** mosse

**FOR** **EACH** **ROW**

**BEGIN**

**if(new.**potenza **>** 250 **OR** **new.**potenza **<** -2**)** **THEN** **SET** **new.**Potenza **=** -2**;**

**end** **if;**

**END**