

BASI DI DATI - Progetto gruppo 22

A.A. 2020/2021

**Realizzato da Francesco Malferrari,
Lucrezia Sitta e Lorenzo Rossi**

Indice

Indice	1
Descrizione dell'idea implementata	3
Glossario	5
Schema scheletro	7
Strategia progettuale	8
PROGETTAZIONE CONCETTUALE	9
Auto-associazione	9
ID esterno	10
Gerarchia	11
Dato derivato	12
Schema E-R completo	15
Progetto logico	16
Eliminazione gerarchie ISA	16
Selezione delle chiavi primarie ed eliminazione degli ID. esterni	17
Trasformazione degli attributi composti o multipli	18
Traduzione di entità e associazioni in schemi di relazioni	20
Verifica di normalizzazione	24
Creazione tabelle	25
Trigger	31
Prevenzione dei cicli evolutivi	31
Deduplicazione evolutiva delle mosse imparate	32
Controllo esclusività dispositivo	33
Aggiornamento dato derivato	34
Controllo integrità del codice fiscale	35
Inserimento dati	37
Studio di alcune query SQL	52

Descrizione dell'idea implementata

Il progetto ha scopo enciclopedico e permette di gestire un servizio per fare in modo che gli utenti possano conoscere gli aspetti più importanti riguardanti l'universo Pokémon. Infatti non è focalizzato su un aspetto in particolare, ma sulla ricerca e l'introduzione di qualsiasi contenuto che abbia qualche collegamento con questo mondo creato dall'azienda giapponese Nintendo, tra cui i videogiochi, gli anime e i manga in cui questi compaiono, le mosse, le evoluzioni e le zone in cui possono essere catturati all'interno dei giochi e altri aspetti che verranno analizzati.

Questo mondo ha come fulcro i Pokémon, mostri tascabili usati spesso in sfide tra allenatori, che sono caratterizzati da un codice identificativo (l'indice del Pokédex nazionale), un nome, una descrizione, in quale zona possono essere catturati e la percentuale del sesso (con cui sono catturabili).

In un'enciclopedia digitale riguardante questo brand non si poteva non citare l'applicazione per IOS e Android "Pokémon Go!" che ha avuto un boom di download nell'estate del 2016 e che tuttora fattura miliardi di dollari all'anno, in particolare nel nostro progetto ci si concentra nell'elencare le varie zone dove ciascun Pokémon può essere catturato all'interno dell'app.

Ogni Pokémon può conoscere un certo numero di mosse, con le varie caratteristiche tra cui il nome, i danni, se è imparabile con una tm e la categoria (se è "fisica", "speciale", ecc...). Ogni mossa fa parte di un tipo ("Normale", "Erba", "Veleno", ecc...) che ha un grado di efficacia contro un altro tipo: per esempio se fuoco combatte contro acqua non è molto efficace, viceversa invece lo è molto.

Alcuni Pokémon possono avere delle evoluzioni, o scatenati dal livello oppure da richieste specifiche. Determinati di questi possono evolversi con più evoluzioni possibili, ma ogni evoluzione ha solo un Pokémon di provenienza (ad esempio Eevee).

Esistono diversi giochi riguardanti il mondo Pokémon dove questi possono comparire, questi sono pubblicati per diverse piattaforme con peculiarità diverse: o come app per smartphone, o come videogioco per una console o come browser game.

Oltre ai giochi, possono comparire in determinati contenuti multimediali, tra cui serie animate, film e manga, e in alcuni casi possono partecipare come personaggi (per esempio Meowth e Pikachu) insieme ai personaggi umani (come Ash e Brock).

Ogni comparsa in una serie o in un film è per forza di cose doppiato: ogni doppiatore può doppiare 1 o più personaggi, possono esistere due doppiatori

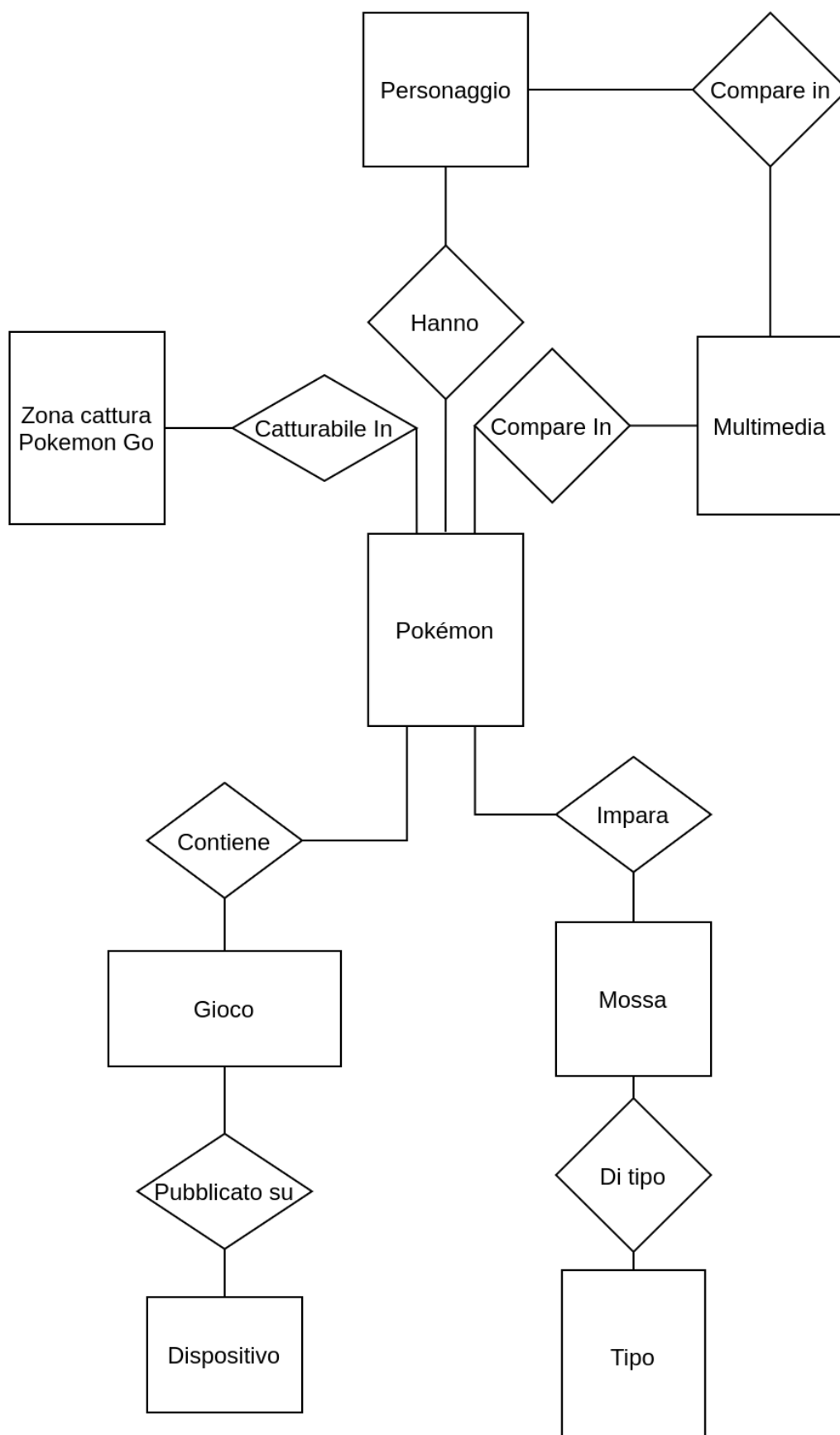
diversi per un personaggio nello stesso multimedia (ad esempio Brock e Meowth) e può esserci colui o colei che doppia lo stesso personaggio in più lingue (ad esempio Pikachu).

Glossario

Termine	Descrizione	Sinonimi	Legame
Gioco	nome, data_rilascio, casa_sviluppo	Videogioco	Dispositivo, Pokémon
Dispositivo	nome	Apparecchio, Strumento	Gioco
Pokémon	id, nome, tipo, descrizione, zona_cattura, percentuale_sesso	Pocket Monster	Gioco, Anime, Zona cattura Pokémon Go, Mossa
Mossa	id, danni, tipo, is_tm		Pokémon
Tipo	nome	TipoMossa	Mossa
Multimedia	id, casa_produzione, anno, regista, tipo_multimediale	Film, Contenuto animato, Anime, Manga	Pokémon
Zona cattura Pokémon Go	stato, città, tipo_zona		Pokémon
Personaggio	nome, cognome, prima_apparizione	Ragazzo, Pokémon, Carattere	Pokémon, Multimedia
Comparsa	personaggio_nome, personaggio_cognome, multimedia_nome, multimedia_anno	Apparizione, Partecipazione	Personaggio, Multimedia, Doppiaggio
Doppiaggio	lingua, personaggio_nome, personaggio_cognome, multimedia_nome, multimedia_anno, doppiatore	Voice acting	Comparsa, Doppiatore
Doppiatore	cod_fis, nome, cognome,	Voice actor, Interprete	Doppiaggio

	data_nascita		
--	--------------	--	--

Schema scheletro



Strategia progettuale

La strategia usata è il Top Down: partendo da uno schema iniziale si è raffinato aggiungendo dettagli e dividendo entità più grandi in più piccole, fino ad arrivare ad uno schema finale.

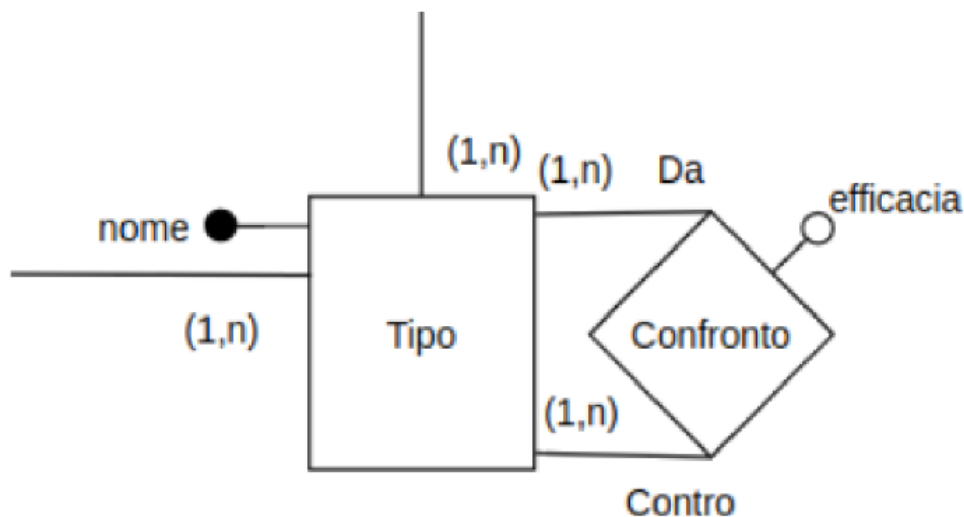
Il procedimento è stato efficace data la facilità nel raffinare una parte di schema senza influenzare le parti restanti, questo grazie alla complessità contenuta dello schema scheletro.

PROGETTAZIONE CONCETTUALE

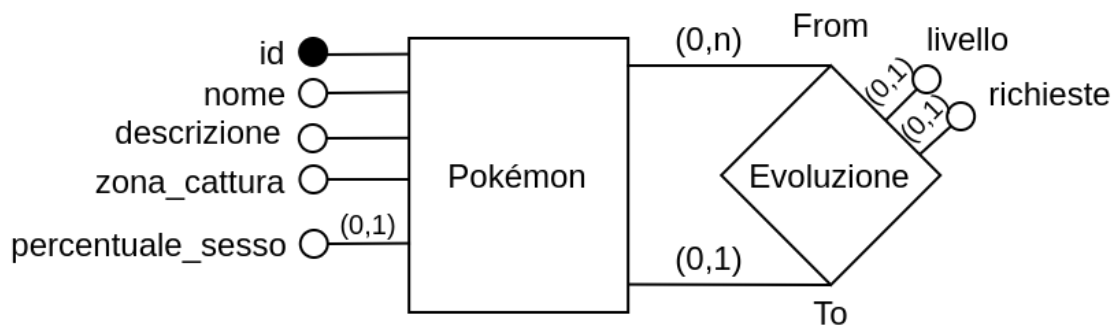
Auto-associazione

L'auto-associazione è stata utilizzata in due contesti diversi:

1) Per confrontare i vari tipi (di Pokémon e di mossa) tra loro, per osservare quanto un tipo (acqua, fuoco, terra, ecc...) è efficace in combattimento contro un altro. Il grado di efficacia è segnato nell'attributo dell'associazione "Confronto".



2) Per evidenziare la scala evolutiva dei Pokémon, dove uno può avere evoluzioni diverse (come nel caso di Eevee) però un'evoluzione ha un solo antenato.

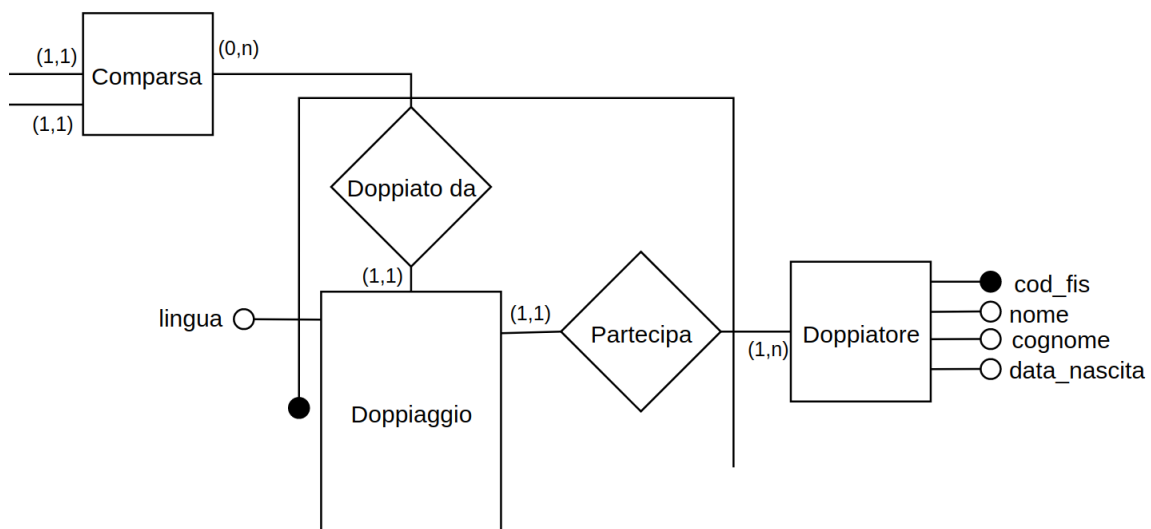


ID esterno

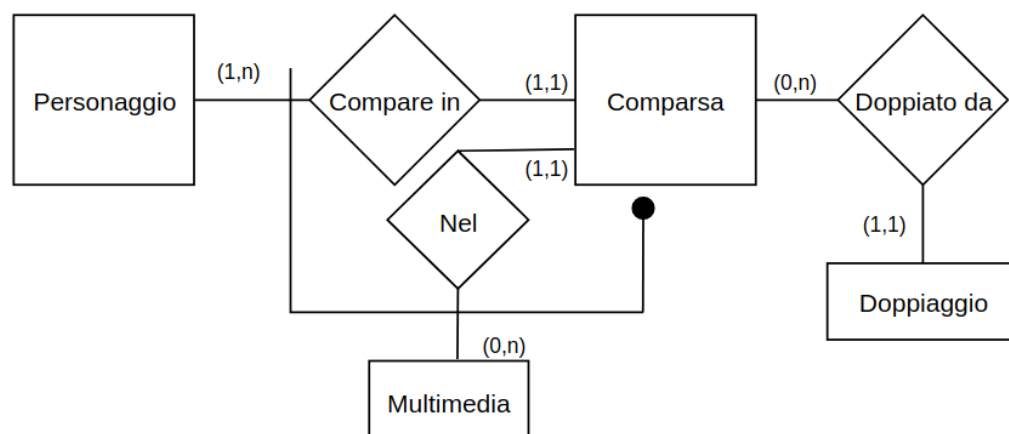
L'identificatore esterno viene usato in due contesti differenti:

1) Per stabilire una dipendenza tra “Doppiaggio” e “Doppiatore”, in modo che non possa esistere un doppiaggio senza quel doppiatore, e tra “Doppiaggio” e “Comparsa”, in modo che esistano solo doppiaggi di personaggi che effettivamente sono presenti nel database.

Anche per fare in modo che un doppiatore possa doppiare uno o più personaggi di uno o più anime, in una o più lingue.

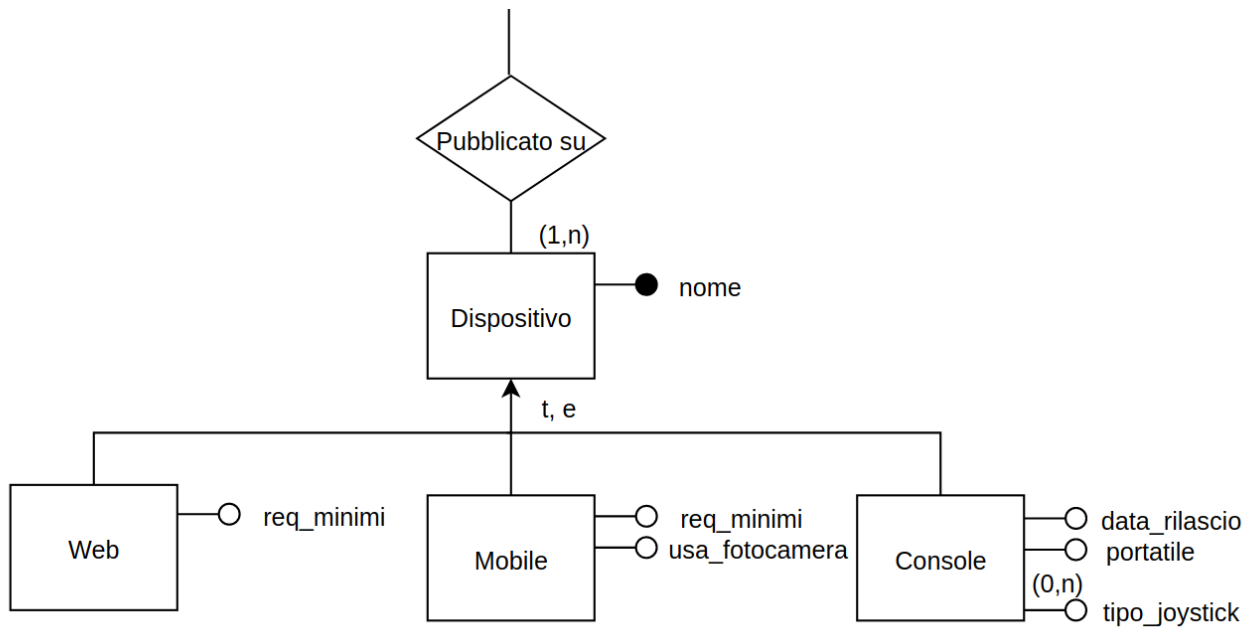


2) Per raggruppare i personaggi nel loro multimedia rispettivo. Non è stata usata una semplice associazione a tre al posto dell'entità perché in tal caso il doppiaggio sarebbe stato incluso nella chiave dell'associazione, quindi non si sarebbe potuto avere personaggi senza doppiaggio o con più doppiaggi.



Gerarchia

La gerarchia nel nostro progetto è stata utilizzata per distinguere i vari dispositivi di supporto di giochi nei quali è presente almeno un Pokémon. La separazione è dovuta a una differenza di caratteristiche tra le piattaforme sia dal punto di vista dell'Hardware, del Software e delle esigenze di gameplay (per esempio fotocamera e joystick).



Dato derivato

Il nostro dato derivato “num_pokemon” serve per contare il numero di Pokémon presenti in ogni gioco direttamente sull’entità “Gioco”.

In questo caso studieremo due operazioni principali:

1. La lettura del numero di Pokémon sapendo il nome del gioco.
2. L’aggiunta di un Pokémon ad un dato gioco (con nome ed id conosciuti).

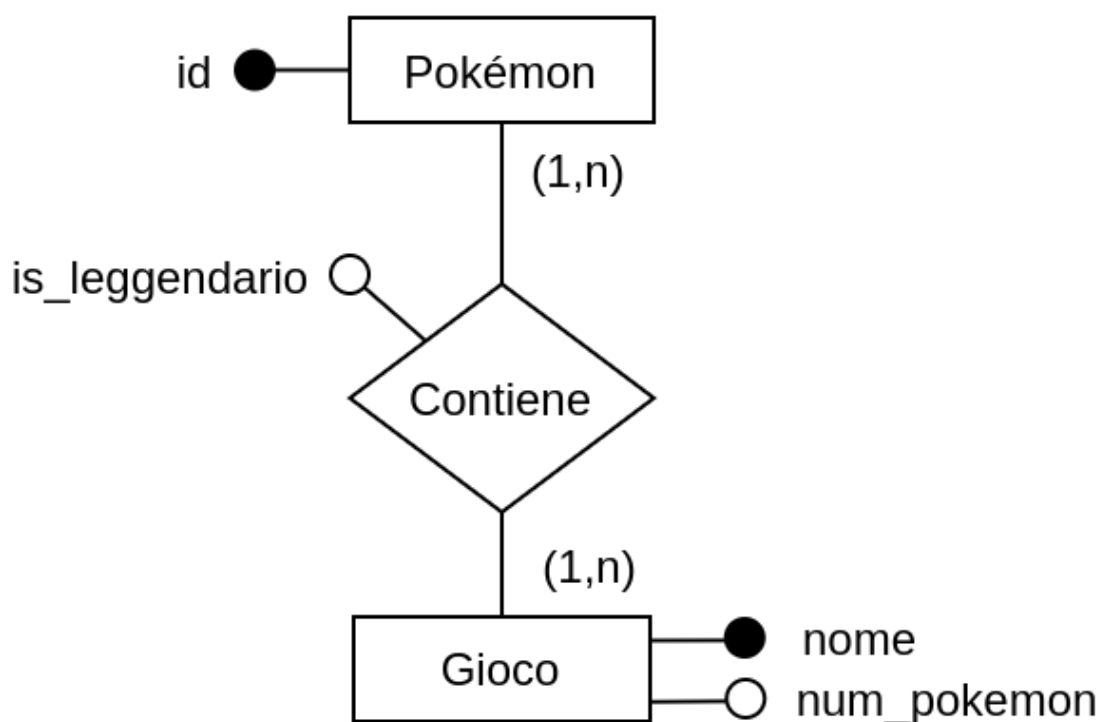


TABELLA DEI VOLUMI

TITOLO	TIPO	VOLUME
Pokémon	E	900
Gioco	E	80
Contiene	R	16000

Stimiamo che per ogni gioco ci sono (in media) 200 Pokémon contenuti.

TABELLA DELLE OPERAZIONI

OPERAZIONE	TIPO	FREQUENZA
Operazione 1	I	100 / giorno
Operazione 2	I	5 / giorno

CON DATO DERIVATO

OPERAZIONE 1

CONCETTO	ACCESSO	TIPO
Gioco	1	Lettura

$$1 * 100/\text{Giorno} = 100/\text{Giorno}$$

OPERAZIONE 2

CONCETTO	ACCESSO	TIPO
Contiene	1	Scrittura
Gioco	1	Lettura
Gioco	1	Scrittura

$$(1 + 2*2) * 5/\text{Giorno} = 25/\text{Giorno}$$

TOTALE

$$100/\text{Giorno} + 25/\text{Giorno} = 125/\text{Giorno}$$

SENZA DATO DERIVATO

OPERAZIONE 1

CONCETTO	ACCESSO	TIPO
Gioco	1	Lettura
Contiene	200	Lettura

$$(200+1) * 100/\text{Giorno} = 20100/\text{Giorno}$$

OPERAZIONE 2

CONCETTO	ACCESSO	TIPO
Contiene	1	Scrittura

$$(1*2) * 5/\text{Giorno} = 10/\text{Giorno}$$

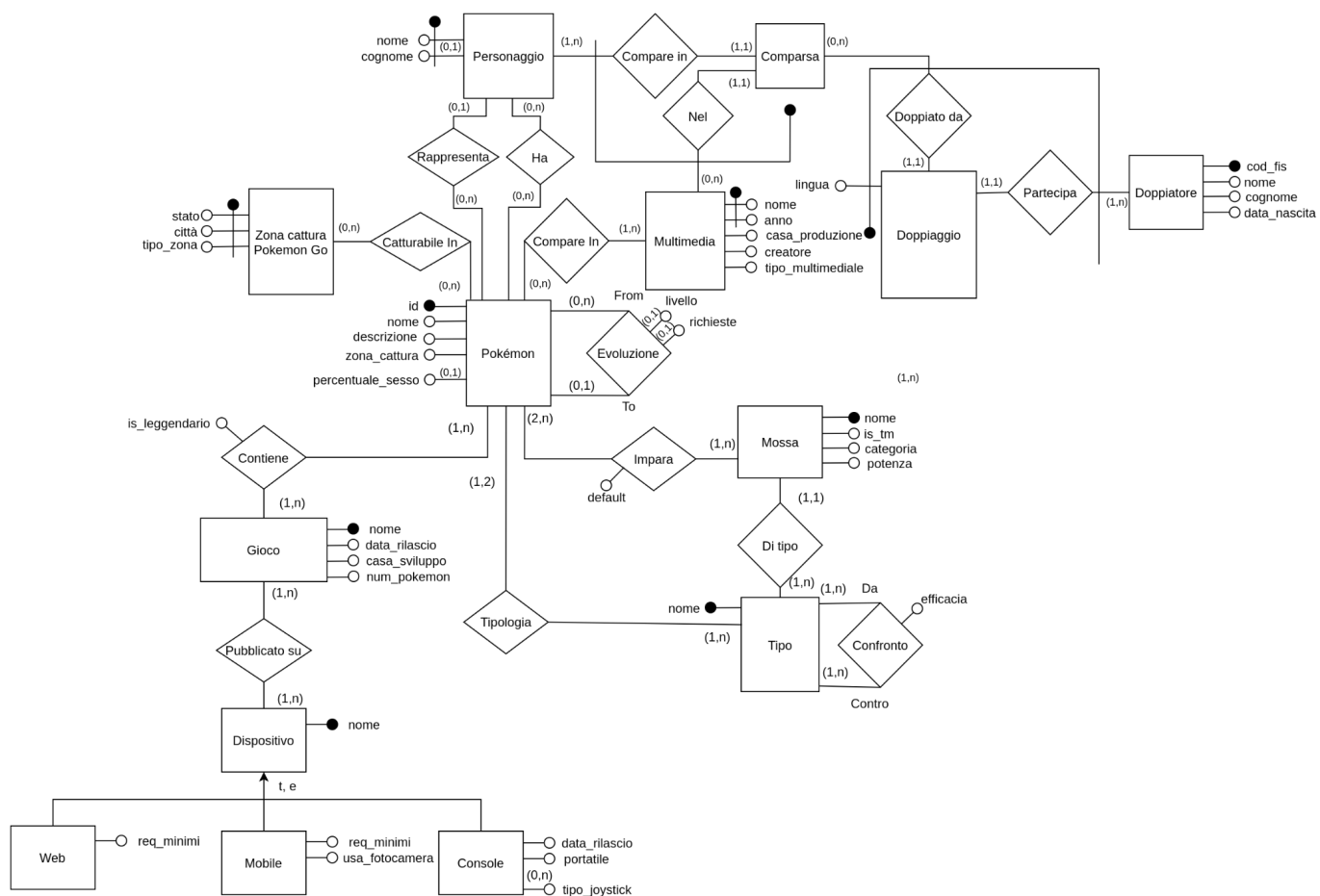
TOTALE

$$20100/\text{Giorno} + 10/\text{Giorno} = 20110/\text{Giorno}$$

CONCLUSIONE

In quanto il nostro database ha un numero di letture di gran lunga superiore al numero di scritture e data la dispendiosità delle letture senza dato derivato, si può calcolare che è più conveniente usare il dato derivato.

Schema E-R completo

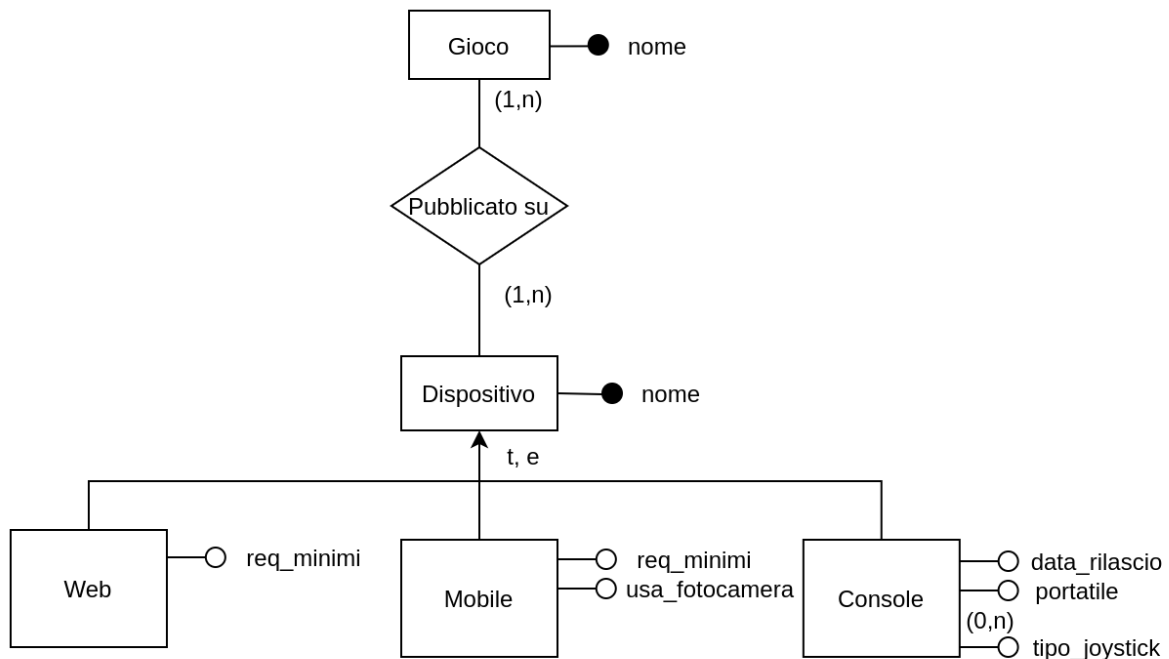


Progetto logico

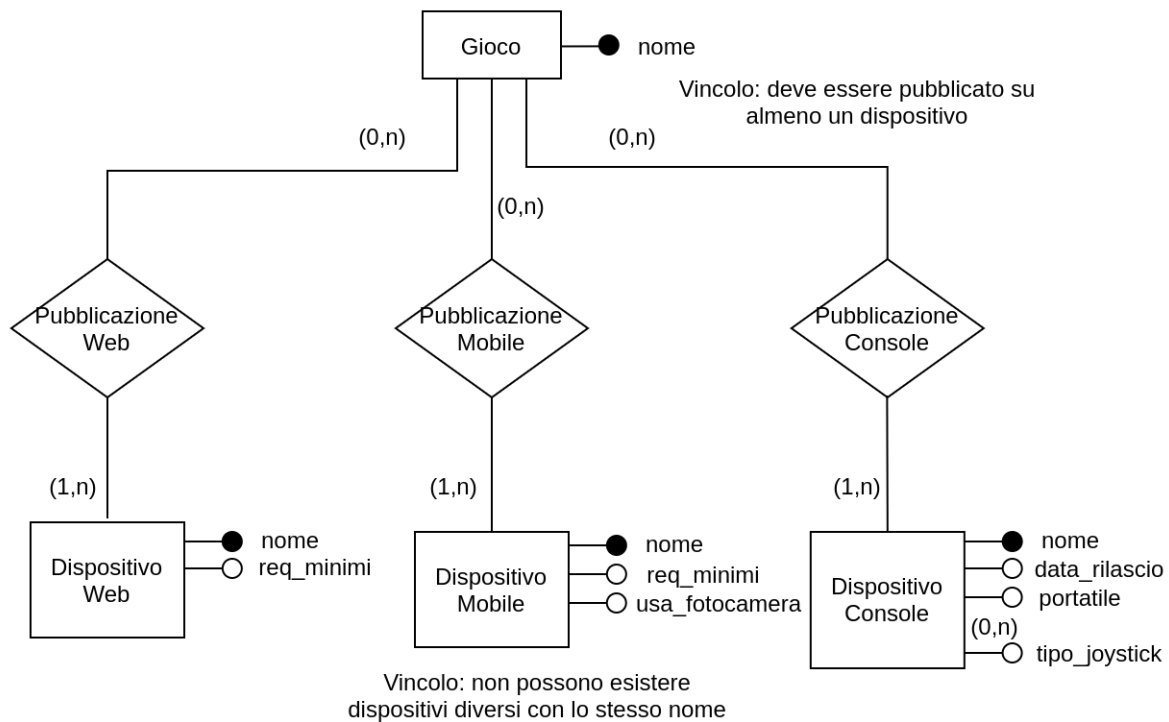
La traduzione dello schema ER in progetto logico è diviso in diversi passi, qui documenteremo i passi usati in questo progetto:

Eliminazione gerarchie ISA

L'unica gerarchia presente nel progetto è quella della tipologia del dispositivo.

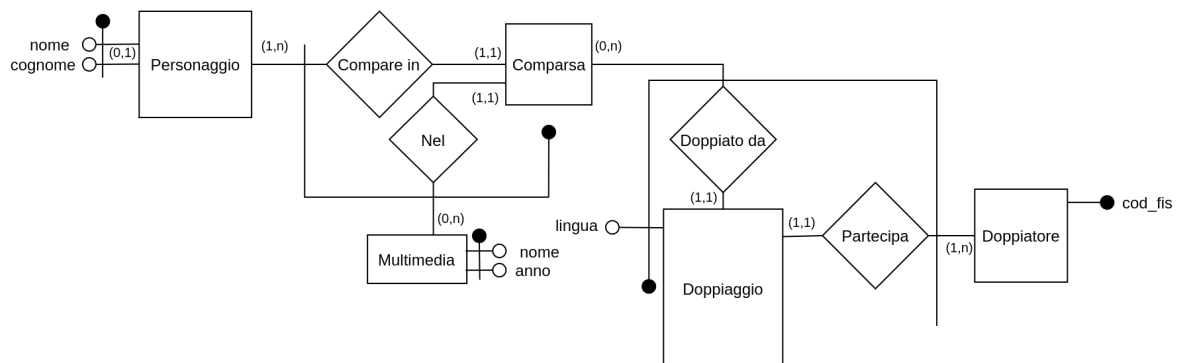


Dato che non ci sono molte informazioni riguardanti al dispositivo padre abbiamo preferito eseguire un collasso verso il basso, tenendo quindi tutte le entità figlie che quindi ereditano sia gli attributi (in questo caso solo il nome) sia le relazioni (la pubblicazione).

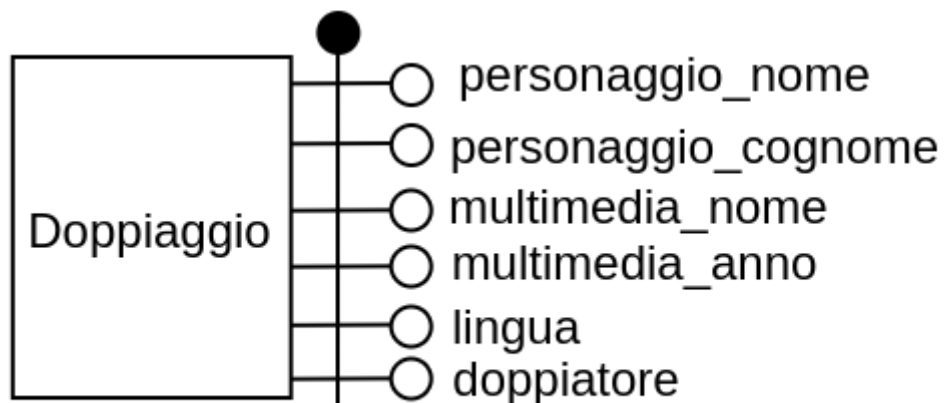
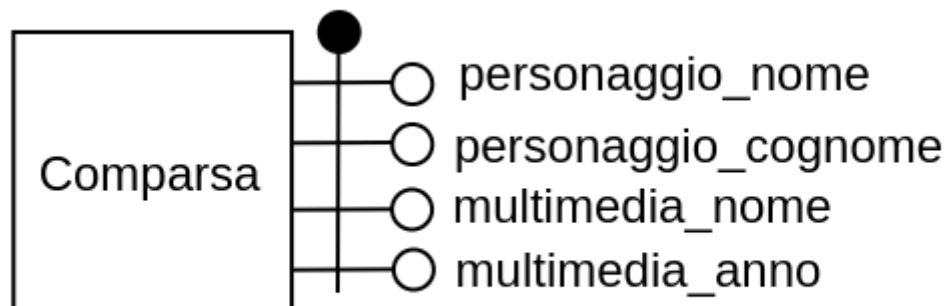


Selezione delle chiavi primarie ed eliminazione degli ID. esterni

Questo passaggio riguarda quindi la parte dei multimedia.

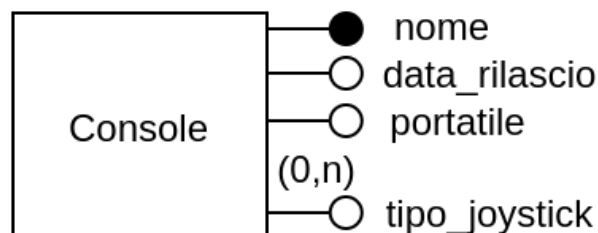


Si può notare che “Doppiaggio” e “Comparsa” hanno chiavi esterne, in questo passaggio si devono eliminare aggiungendo gli identificatori delle entità esterne all’entità (mettendole in chiave). Qua sotto verranno mostrate solo le immagini delle entità “Comparsa” e “Doppiaggio” dato che le altre sono rimaste invariate.

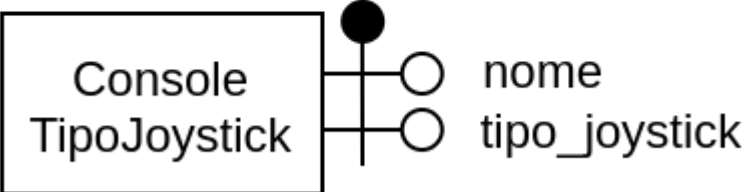
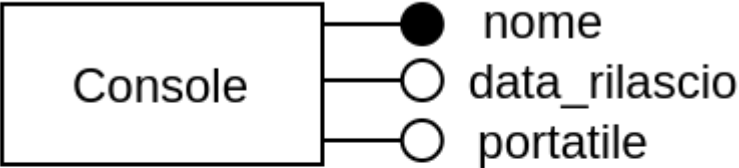


Trasformazione degli attributi composti o multipli

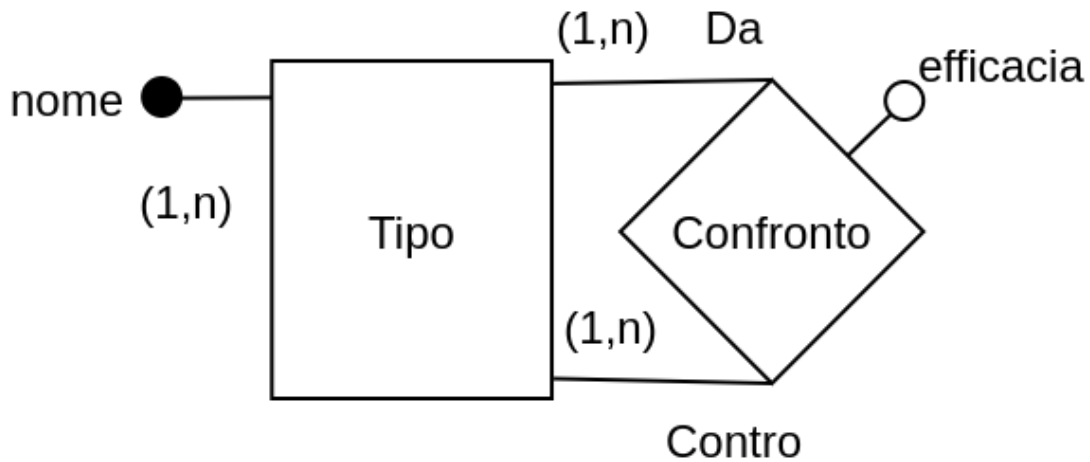
Il progetto non usa nessun attributo composto ed ha solamente un attributo multiplo, presente nell'entità Console.



Questo si può trasformare semplicemente creando un'altra entità.



Traduzione di entità e associazioni in schemi di relazioni



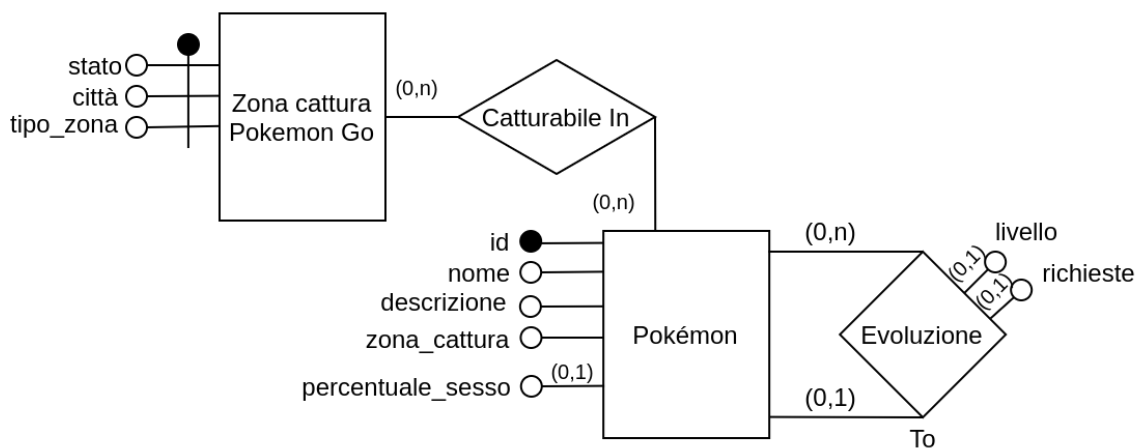
Tipo(nome)

ConfrontoTipo(da, contro, efficacia)

FK: da REFERENCES Tipo(nome)

FK: contro REFERENCES Tipo(nome)

Vincolo: efficacia tra 0, 1, 2



Pokemon(id, nome, descrizione, zona_cattura, percentuale_sesso, tipo1, tipo2)

FK: tipo1 REFERENCES Tipo(nome)

FK: tipo2 REFERENCES Tipo(nome)

Vincolo: tipo2 se presente deve essere diverso da tipo1

Vincolo: zona_cattura tra ["Foresta", "Spiaggia", "Caverna", "Evento", "Erba alta"]

Evoluzione(to, from, livello, richieste)

FK: to REFERENCES Pokemon(id)

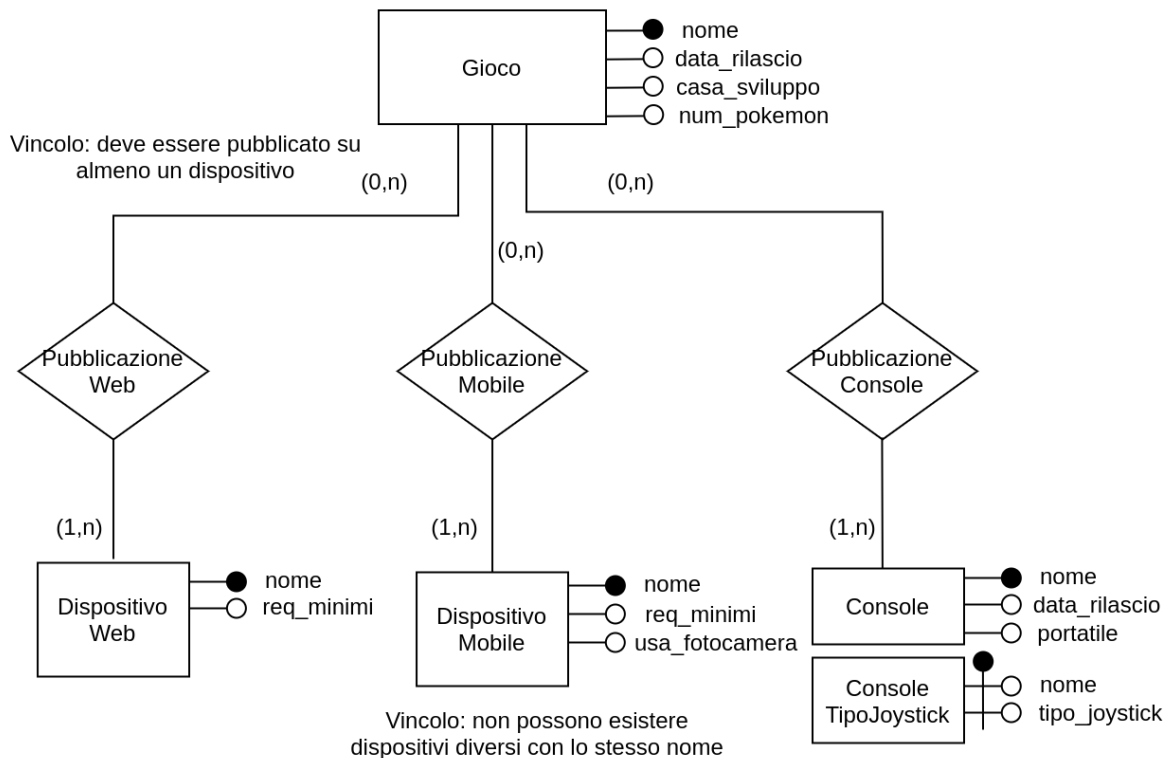
FK: from REFERENCES Pokemon(id)

Vincolo: non ci devono essere cicli evolutivi

ZonaCatturaPokemonGo(stato, citta, tipo_zona, pokemon)

FK: pokemon REFERENCES Pokemon(id)

Vincolo: tipo_zona tra ["Mare", "Urbano", "Rurale"]



Gioco(nome, data_rilascio, casa_sviluppo, num_pokemon)

Vincolo: num_pokemon deve rappresentare il numero di pokemon contenuti in quel gioco

DispositivoWeb(nome, req_minimi)

DispositivoMobile(nome, req_minimi, usa_fotocamera)

DispositivoConsole(nome, data_rilascio, portatile)

Vincolo: ogni nome deve essere presente al massimo in un dispositivo.

ConsoleTipoJoystick(nome, tipo_joystick)

FK: nome REFERENCES DispositivoConsole(nome)

Commento: tipo_joystick non ha nessun vincolo sul valore per dare la possibilità di aggiungere nuovi joystick

PubblicazioneWeb(gioco, dispositivo)

FK: gioco REFERENCES Gioco(nome)

FK: dispositivo REFERENCES DispositivoWeb(nome)

PubblicazioneMobile(gioco, dispositivo)

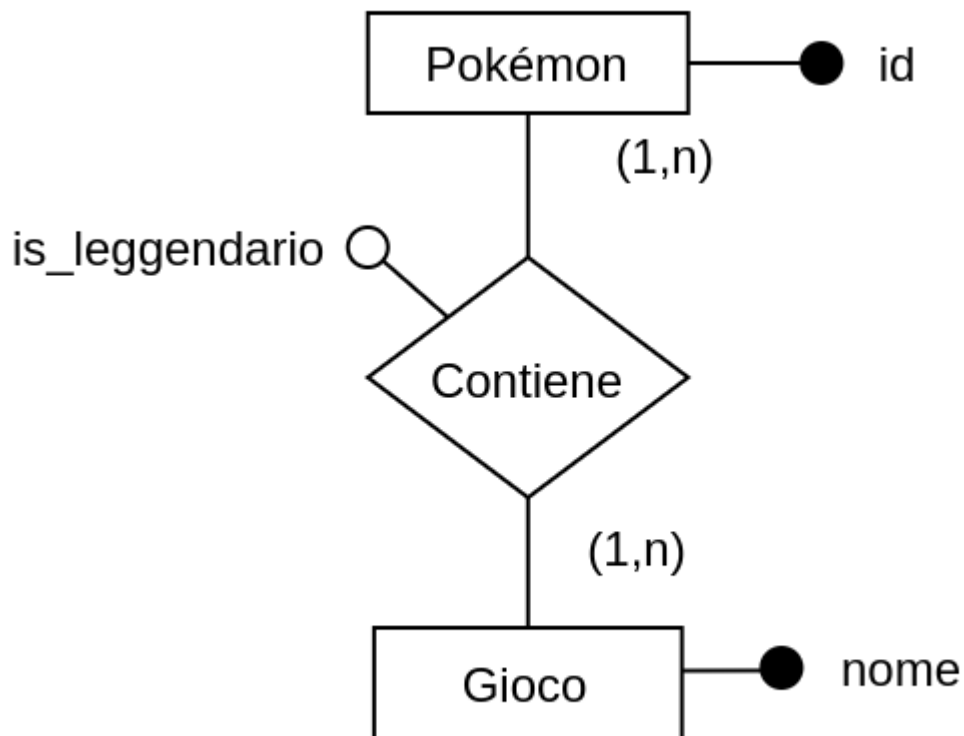
FK: gioco REFERENCES Gioco(nome)

FK: dispositivo REFERENCES DispositivoMobile(nome)

PublicazioneConsole(gioco, dispositivo)

FK: gioco REFERENCES Gioco(nome)

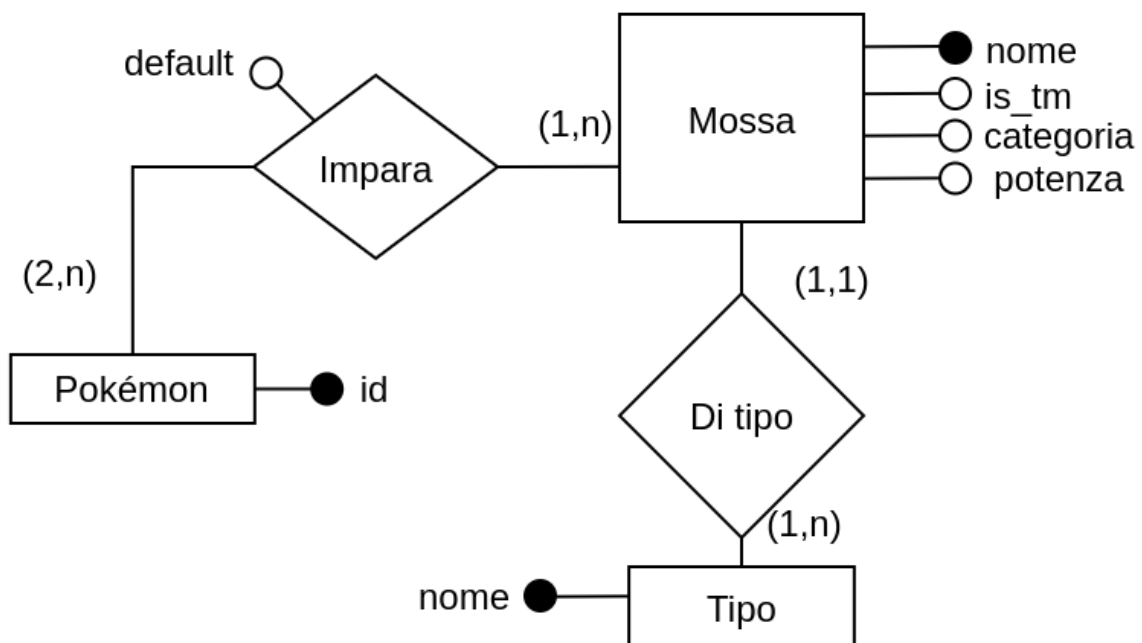
FK: dispositivo REFERENCES DispositivoConsole(nome)



PokemonInGioco(pokemon, gioco, is_leggendario)

FK: gioco REFERENCES Gioco(nome)

FK: pokemon REFERENCES Pokemon(id)



Mossa(nome, tipo, is_tm, categoria, potenza)

FK: tipo REFERENCES Tipo(nome)

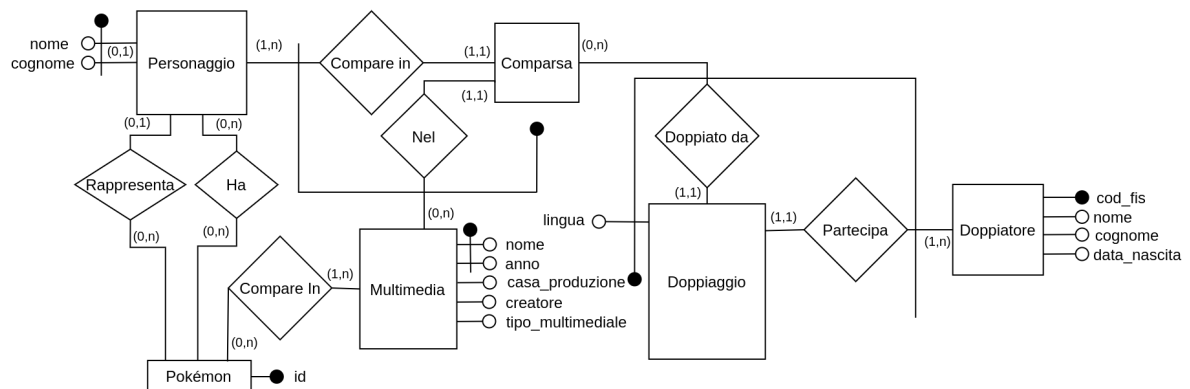
Vincolo: categoria tra ["Fisica", "Speciale", "Stato"]

ImparaMossa(pokemon, mossa, default)

FK: pokemon REFERENCES Pokemon(id)

FK: mossa REFERENCES Mossa(nome)

Vincolo: se una mossa è imparata da un pokemon non deve essere inserita nelle sue evoluzioni (per evitare la duplicazione di dati)



Personaggio(nome, cognome, rappresenta_pokemon)

FK: rappresenta_pokemon REFERENCES Pokemon(id)

Commento: se il cognome non esiste o è sconosciuto si sostituisce con una stringa nulla

PokemonPersonaggio(personaggio_nome, personaggio_cognome, pokemon)

FK: (personaggio_nome, personaggio_cognome) REFERENCES

Personaggio(nome, cognome)

FK: pokemon REFERENCES Pokemon(id)

Multimedia(nome, anno, casa_produzione, creatore, tipo_multimediale)

Vincolo: tipo_multimediale tra ["Manga", "Serie anime", "Film anime", "Film live action"]

CompareIn(pokemon, multimedia_nome, multimedia_anno)

FK: pokemon REFERENCES Pokemon(id)

FK: (multimedia_nome, multimedia_anno) REFERENCES

Multimedia(nome, anno)

Comparsa(personaggio_nome, personaggio_cognome, multimedia_nome, multimedia_anno)

FK: (personaggio_nome, personaggio_cognome) REFERENCES

Personaggio(nome, cognome)

FK: (multimedia_nome, multimedia_anno) REFERENCES Multimedia(nome,

anno)

Doppiatore(cod_fis, nome, cognome, data_nascita)

Doppiaggio(personaggio_nome, personaggio_cognome, multimedia, lingua,
doppiatore)

FK: (personaggio_nome, personaggio_cognome, multimedia)

REFERENCES Comparsa(personaggio_nome, personaggio_cognome,
multimedia)

FK: doppiatore REFERENCES Doppiatore(cod_fis)

Verifica di normalizzazione

Le relazioni risultano in forma naturale dato che lo schema ER era ben formato.

Creazione tabelle

```
CREATE TABLE tipo (  
    nome varchar(20),  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE Pokemon (  
    id int check (id > 0),  
    nome varchar(20) NOT NULL,  
    descrizione varchar(300),  
    zona_cattura varchar(20) check (zona_cattura in ('Foresta',  
'Spiaggia', 'Caverna', 'Evento', 'Erba alta')) NOT NULL,  
    percentuale_sesso float check (percentuale_sesso <= 100 AND  
percentuale_sesso >= 0),  
    tipo1 varchar(20) NOT NULL,  
    tipo2 varchar(20) check (tipo2 <> tipo1),  
    PRIMARY KEY (id),  
    FOREIGN KEY (tipo1) REFERENCES Tipo(nome),  
    FOREIGN KEY (tipo2) REFERENCES Tipo(nome)  
);
```

```
CREATE TABLE Evoluzione (  
    da int,  
    a int,  
    livello int,  
    richieste varchar(100),  
    PRIMARY KEY (a),  
    FOREIGN KEY (da) REFERENCES pokemon(id),  
    FOREIGN KEY (a) REFERENCES pokemon(id)  
);
```

```
CREATE TABLE ZonaCatturaPokemonGo (  
    stato varchar(30),  
    citta varchar(30),  
    tipo_zona varchar(30) check (tipo_zona in  
('Mare', 'Urbano', 'Rurale')),
```

```
    pokemon int NOT NULL,  
    PRIMARY KEY(stato,citta,tipo_zona,pokemon),  
    FOREIGN KEY (pokemon) REFERENCES pokemon(id)  
);
```

```
CREATE TABLE gioco (  
    nome varchar(50),  
    num_pokemon int NOT NULL DEFAULT 0,  
    data_rilascio date,  
    casa_sviluppo varchar(20) NOT NULL,  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE dispositivoweb (  
    nome varchar(20),  
    req_minimi varchar(100),  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE dispositivomobile (  
    nome varchar(20),  
    req_minimi varchar(100),  
    usa_fotocamera boolean,  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE dispositivoconsole (  
    nome varchar(20),  
    data_rilascio date,  
    portatile boolean,  
    PRIMARY KEY (nome)  
);
```

```
CREATE TABLE consoletipojoystick (  
    nome varchar(20),  
    tipo_joystick varchar(20) NOT NULL,  
    PRIMARY KEY (nome),  
    FOREIGN KEY (nome) REFERENCES dispositivoconsole(nome)
```

);

```
CREATE TABLE pubblicazioneweb (  
    gioco varchar(50),  
    dispositivo varchar(20),  
    PRIMARY KEY (gioco,dispositivo),  
    FOREIGN KEY (gioco) REFERENCES gioco(nome),  
    FOREIGN KEY (dispositivo) REFERENCES dispositivoweb(nome)  
);
```

```
CREATE TABLE pubblicazionemobile (  
    gioco varchar(50),  
    dispositivo varchar(20),  
    PRIMARY KEY (gioco,dispositivo),  
    FOREIGN KEY (gioco) REFERENCES gioco(nome),  
    FOREIGN KEY (dispositivo) REFERENCES dispositivomobile(nome)  
);
```

```
CREATE TABLE pubblicazioneconsole (  
    gioco varchar(50),  
    dispositivo varchar(20),  
    PRIMARY KEY (gioco,dispositivo),  
    FOREIGN KEY (gioco) REFERENCES gioco(nome),  
    FOREIGN KEY (dispositivo) REFERENCES  
dispositivoconsole(nome)  
);
```

```
CREATE TABLE pokemoningioco (  
    pokemon int,  
    gioco varchar(50),  
    is_leggendario boolean NOT NULL,  
    PRIMARY KEY (pokemon, gioco),  
    FOREIGN KEY (gioco) REFERENCES gioco(nome),  
    FOREIGN KEY (pokemon) REFERENCES pokemon(id)  
);
```

```
CREATE TABLE ConfrontoTipo (  
    da varchar(20),
```

```

        contro varchar(20),
        efficacia int check (efficacia in (0,1,2)) NOT NULL,
        PRIMARY KEY (da,contro),
        FOREIGN KEY (da) REFERENCES tipo(nome),
        FOREIGN KEY (contro) REFERENCES tipo(nome)
    );

```

```

CREATE TABLE mossa (
    nome varchar(20),
    tipo varchar(20) NOT NULL,
    is_tm boolean NOT NULL,
    categoria varchar(10) check (categoria in
('Fisica','Speciale','Stato')) NOT NULL,
    potenza int NOT NULL,
    PRIMARY KEY (nome),
    FOREIGN KEY (tipo) REFERENCES tipo(nome)
);

```

```

CREATE TABLE imparamossa (
    pokemon int,
    mossa varchar(20),
    partenza boolean NOT NULL,
    PRIMARY KEY (pokemon,mossa),
    FOREIGN KEY (pokemon) REFERENCES pokemon(id),
    FOREIGN KEY (mossa) REFERENCES mossa(nome)
);

```

```

CREATE TABLE personaggio (
    nome varchar(20),
    cognome varchar(20),
    rappresenta_pokemon int,
    PRIMARY KEY (nome,cognome),
    FOREIGN KEY (rappresenta_pokemon) REFERENCES pokemon(id)
);

```

```

CREATE TABLE pokemonpossesso (
    personaggio_nome varchar(20),
    personaggio_cognome varchar(20),

```

```

    pokemon int,
    PRIMARY KEY (personaggio_nome,personaggio_cognome,pokemon),
    FOREIGN KEY (personaggio_nome,personaggio_cognome)
REFERENCES personaggio(nome,cognome),
    FOREIGN KEY (pokemon) REFERENCES pokemon(id)
);

```

```

CREATE TABLE multimedia (
    nome varchar(40),
    anno int check (anno > 1980),
    casa_produzione varchar(40) NOT NULL,
    creatore varchar(40),
    tipo_multimediale varchar(20) check (tipo_multimediale in
('Manga','Serie anime','Film anime','Film live action')) NOT
NULL,
    PRIMARY KEY (nome,anno)
);

```

```

CREATE TABLE comparein (
    pokemon int,
    multimedia_nome varchar(40),
    multimedia_anno int,
    PRIMARY KEY (pokemon,multimedia_nome,multimedia_anno),
    FOREIGN KEY (pokemon) REFERENCES pokemon(id),
    FOREIGN KEY (multimedia_nome,multimedia_anno) REFERENCES
multimedia(nome,anno)
);

```

```

CREATE TABLE comparsa (
    personaggio_nome varchar(20),
    personaggio_cognome varchar(20),
    multimedia_nome varchar(40),
    multimedia_anno int,
    PRIMARY
KEY(personaggio_nome,personaggio_cognome,multimedia_nome,multi
media_anno),
    FOREIGN KEY (personaggio_nome,personaggio_cognome)
REFERENCES personaggio(nome,cognome),

```

```
    FOREIGN KEY (multimedia_nome,multimedia_anno) REFERENCES  
multimedia(nome,anno)  
);
```

```
CREATE TABLE doppiatore (  
    cod_fis char(16),  
    nome varchar(20) NOT NULL,  
    cognome varchar(20) NOT NULL,  
    data_nascita date,  
    PRIMARY KEY (cod_fis)  
);
```

```
CREATE TABLE doppiaggio (  
    personaggio_nome varchar(20),  
    personaggio_cognome varchar(20),  
    multimedia_nome varchar(40),  
    multimedia_anno int,  
    lingua varchar(20),  
    doppiatore char(16),  
    PRIMARY  
KEY(personaggio_nome,personaggio_cognome,multimedia_nome,multi  
media_anno,lingua,doppiatore),  
    FOREIGN KEY  
(personaggio_nome,personaggio_cognome,multimedia_nome,multimed  
ia_anno) REFERENCES  
comparsa(personaggio_nome,personaggio_cognome,multimedia_nome,  
multimedia_anno),  
    FOREIGN KEY (doppiatore) REFERENCES doppiatore(cod_fis)  
);
```

Trigger

Prevenzione dei cicli evolutivi

I Pokémon non possono avere cicli nelle evoluzioni (A non può evolversi in B se B si evolve in A). Abbiamo deciso di implementare questa prevenzione perché è indispensabile, la stessa tecnica verrà usata nel prossimo trigger per prevenire la duplicazione dei dati.

```
CREATE OR REPLACE FUNCTION check_evolution_cycle() RETURNS
trigger AS $$
BEGIN
    IF (TG_OP = 'INSERT' OR NEW.da <> OLD.da OR NEW.a <>
OLD.a) AND EXISTS (
        WITH RECURSIVE search_graph(current, cycle) AS (
            SELECT NEW.da, (NEW.da = NEW.a)
            UNION ALL
                SELECT g.da, g.da = NEW.a
                FROM evoluzione g, search_graph sg
                WHERE g.a = sg.current AND NOT cycle
        )
        SELECT * FROM search_graph
            WHERE cycle
            LIMIT 1
    )
    THEN
        RAISE EXCEPTION 'Detected cycle in evolution';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER detect_evolution_cycle_after_update
AFTER INSERT OR UPDATE ON evoluzione
FOR EACH ROW EXECUTE PROCEDURE check_evolution_cycle();
```

Deduplicazione evolutiva delle mosse imparate

Le mosse imparate da un Pokémon si mantengono anche nelle sue evoluzioni successive, al posto di duplicarle per ogni evoluzione si è deciso di andare a cercare quelle imparabili dal Pokémon anche nei suoi antenati evolutivi. Questo trigger si assicura quindi che non ci siano mosse duplicate.

```
CREATE OR REPLACE FUNCTION check_impara_mossa_evolution()
RETURNS trigger AS $$
BEGIN
    IF EXISTS (
        WITH RECURSIVE search_graph(current, duplication) AS (
            SELECT NEW.pokemon, false
            UNION ALL
                SELECT g.da, EXISTS (
                    SELECT * FROM imparamossa WHERE pokemon =
g.da AND mosca = NEW.mossa)
                FROM evoluzione g, search_graph sg
                WHERE g.a = sg.current AND NOT duplication
        )
        SELECT * FROM search_graph
            WHERE duplication
            LIMIT 1
    )
    THEN
        RAISE EXCEPTION 'Move is already present in previous
evolution';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER dedup_impara_mossa_evolution
AFTER INSERT OR UPDATE ON imparamossa
FOR EACH ROW EXECUTE PROCEDURE check_impara_mossa_evolution();
```


Controllo esclusività dispositivo

Il collasso verso il basso applicato allo schema ER ha rimosso l'esclusività implicita del dispositivo, questo significa che possono esistere due dispositivi di tipo diverso con lo stesso nome (dato che sono tabelle separate), questo trigger si assicura l'unicità del nome.

```
CREATE OR REPLACE FUNCTION check_device_unique() RETURNS
trigger AS $$
BEGIN
    IF (TG_OP = 'INSERT' OR OLD.nome <> NEW.nome) AND 1 < (
SELECT SUM(tbl.cnt) FROM (
        SELECT COUNT(*) as cnt FROM dispositivoweb WHERE
nome = NEW.nome UNION ALL
        SELECT COUNT(*) as cnt FROM dispositivomobile WHERE
nome = NEW.nome UNION ALL
        SELECT COUNT(*) as cnt FROM dispositivoconsole WHERE
nome = NEW.nome
    ) tbl)
    THEN
        RAISE EXCEPTION 'Duplicated device found';
    END IF;
    RETURN NEW;
END
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER detect_device_duplication_web
AFTER INSERT OR UPDATE ON dispositivoweb
FOR EACH ROW EXECUTE PROCEDURE check_device_unique();
```

```
CREATE TRIGGER detect_device_duplication_mobile
AFTER INSERT OR UPDATE ON dispositivomobile
FOR EACH ROW EXECUTE PROCEDURE check_device_unique();
```

```
CREATE TRIGGER detect_device_duplication_console
AFTER INSERT OR UPDATE ON dispositivoconsole
FOR EACH ROW EXECUTE PROCEDURE check_device_unique();
```

Aggiornamento dato derivato

Il dato derivato “num_pokemon” conta il numero di Pokémon presenti in un dato gioco, questo però dev’essere aggiornato.

```
CREATE OR REPLACE FUNCTION update_gioco_pokemon_count()
RETURNS trigger AS $$
BEGIN
    IF (TG_OP = 'DELETE' OR (TG_OP = 'UPDATE' AND NEW.gioco
<> OLD.gioco))
    THEN
        UPDATE gioco SET num_pokemon = num_pokemon - 1 WHERE nome
= OLD.gioco;
    END IF;
    IF (TG_OP = 'INSERT' OR (TG_OP = 'UPDATE' AND NEW.gioco
<> OLD.gioco))
    THEN
        UPDATE gioco SET num_pokemon = num_pokemon + 1 WHERE nome
= NEW.gioco;
    END IF;
    RETURN NULL; -- We're after the action, return type is
not checked
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER update_gioco_pokemon_count_trigger
AFTER INSERT OR UPDATE OR DELETE ON pokemoningioco
FOR EACH ROW EXECUTE PROCEDURE update_gioco_pokemon_count();
```

Controllo integrità del codice fiscale

L'ultimo carattere del codice fiscale viene calcolato in base ad un semplice algoritmo e serve alle macchine per rilevare errori nell'inserimento del codice fiscale. Questo check è più che altro un modo per esplorare plpgsql come linguaggio di programmazione generico.

```
CREATE OR REPLACE FUNCTION
check_codice_fiscale_carattere_di_controllo(cod CHAR(16))
RETURNS BOOLEAN AS $$
DECLARE
    -- -1 is for filling since ASCII has spaces between '9'
    and 'A'
    -- Remember that PostgreSQL arrays are 1-based
    table1 constant integer[] := ARRAY [1, 0, 5, 7, 9, 13,
15, 17, 19, 21, -1, -1, -1, -1, -1, -1, -1, 1, 0, 5, 7, 9, 13,
15, 17, 19, 21, 2, 4, 18, 20, 11, 3, 6, 8, 12, 14, 16,
10, 22, 25, 24, 23];
    table2 constant integer[] := ARRAY [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, -1, -1, -1, -1, -1, -1, -1, 0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25];
    i integer := 0;
    s integer := 0;
    ch char(1) := ' ';
BEGIN
    IF char_length(cod) <> 16 THEN
        RETURN false;
    END IF;

    FOREACH ch IN ARRAY regexp_split_to_array(cod, '')
    LOOP
        IF i = 15 THEN
            RETURN ch = chr((s % 26) + ascii('A'));
        ELSIF (i & 1) = 0 THEN
            s := s + table1[ascii(ch) - ascii('0') + 1];
        ELSE
            s := s + table2[ascii(ch) - ascii('0') + 1];
        END IF;
        i := i + 1;
    END LOOP;
END;
```

```
        END IF;
        i := i + 1;
    END LOOP;

    RETURN NULL;
END
$$ LANGUAGE plpgsql;

ALTER TABLE doppiatore
ADD CONSTRAINT cod_fis_check
    CHECK (cod_fis ~
'^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$\ ' AND
check_codice_fiscale_carattere_di_controllo(cod_fis));
```

Inserimento dati

```
INSERT INTO tipo(nome) VALUES ('normale'), ('fuoco'),  
('acqua'), ('elettro'), ('erba'), ('ghiaccio'), ('lotta'),  
('veleno'), ('terra'), ('volante'), ('psico'), ('coleottero'),  
('roccia'), ('spettro'), ('drago'), ('buio'), ('acciaio');
```

-- 1 -> not effective, 2 -> very effective, 0 -> no damage

```
INSERT INTO confrontotipo(da, contro, efficacia) VALUES
```

```
    ('normale', 'roccia', 1),  
    ('fuoco', 'fuoco', 1),  
    ('fuoco', 'acqua', 1),  
    ('fuoco', 'erba', 2),  
    ('fuoco', 'ghiaccio', 2),  
    ('fuoco', 'roccia', 1),  
    ('acqua', 'fuoco', 2),  
    ('acqua', 'acqua', 1),  
    ('acqua', 'erba', 1),  
    ('acqua', 'terra', 2),  
    ('acqua', 'roccia', 2),  
    ('elettro', 'acqua', 2),  
    ('elettro', 'elettro', 1),  
    ('elettro', 'erba', 1),  
    ('elettro', 'terra', 0),  
    ('elettro', 'volante', 2),  
    ('erba', 'fuoco', 1),  
    ('erba', 'acqua', 2),  
    ('erba', 'erba', 1),  
    ('erba', 'veleno', 1),  
    ('erba', 'terra', 2),  
    ('erba', 'volante', 1),  
    ('erba', 'roccia', 2),  
    ('ghiaccio', 'fuoco', 1),  
    ('ghiaccio', 'acqua', 1),  
    ('ghiaccio', 'erba', 2),  
    ('ghiaccio', 'ghiaccio', 1),  
    ('ghiaccio', 'terra', 2),
```

```

('ghiaccio', 'volante', 2),
('lotta', 'normale', 2),
('lotta', 'erba', 2),
('lotta', 'veleno', 1),
('lotta', 'volante', 1),
('lotta', 'psico', 1),
('lotta', 'roccia', 2),
('lotta', 'buio', 2),
('veleno', 'erba', 2),
('veleno', 'veleno', 1),
('veleno', 'terra', 1),
('veleno', 'roccia', 1),
('terra', 'fuoco', 2),
('terra', 'elettro', 2),
('terra', 'erba', 1),
('terra', 'veleno', 2),
('terra', 'volante', 0),
('terra', 'roccia', 2),
('volante', 'elettro', 1),
('volante', 'erba', 2),
('volante', 'lotta', 2),
('volante', 'roccia', 1),
('psico', 'lotta', 2),
('psico', 'veleno', 2),
('psico', 'psico', 1),
('psico', 'buio', 0),
('roccia', 'fuoco', 2),
('roccia', 'ghiaccio', 2),
('roccia', 'lotta', 1),
('roccia', 'terra', 1),
('roccia', 'volante', 2),
('buio', 'lotta', 1),
('buio', 'psico', 2),
('buio', 'buio', 1);

```

```

INSERT INTO pokemon (id, nome, descrizione, zona_cattura,
percentuale_sesso, tipo1, tipo2) VALUES

```

(1, 'Bulbasaur', 'Fin dalla nascita questo Pokémon ha sulla schiena un seme che cresce lentamente', 'Evento', 87.5, 'erba', 'veleno'),

(2, 'Ivysaur', 'Il bulbo sulla schiena è cresciuto così tanto da impedirgli di alzarsi in piedi sulle zampe posteriori', 'Evento', 87.5, 'erba', 'veleno'),

(3, 'Venusaur', 'Il fiore sboccia assorbendo energia solare. Si muove continuamente in cerca di luce', 'Evento', 87.5, 'erba', 'veleno'),

(4, 'Charmander', 'Ama le cose calde. Si dice che quando piove gli esca vapore dalla punta della coda', 'Evento', 87.5, 'fuoco', NULL),

(5, 'Charmeleon', 'Ha un''indole feroce. Usa la coda fiammeggiante come una frusta e lacera l''avversario con gli artigli affilati.', 'Evento', 87.5, 'fuoco', NULL),

(6, 'Charizard', 'Sputa fiamme incandescenti in grado di fondere le rocce. A volte causa incendi boschivi', 'Evento', 87.5, 'fuoco', 'volante'),

(7, 'Squirtle', 'Quando ritrae il lungo collo dentro la corazza sputa un vigoroso getto d''acqua', 'Evento', 87.5, 'acqua', NULL),

(8, 'Wartortle', 'È considerato un simbolo di longevità. Se c''è del muschio sul suo guscio, significa che è molto anziano', 'Evento', 87.5, 'acqua', NULL),

(9, 'Blastoise', 'Mette KO gli avversari schiacciandoli sotto il corpo possente. Se è in difficoltà, può ritirarsi nella corazza', 'Evento', 87.5, 'acqua', NULL),

(25, 'Pikachu', 'Più potente è l''energia elettrica prodotta dal Pokémon, più le sacche sulle sue guance sono morbide ed elastiche', 'Foresta', 50, 'elettro', NULL),

(26, 'Raichu', 'La sua lunga coda serve da messa a terra per proteggerlo dalla sua stessa alta tensione', 'Foresta', 50, 'elettro', NULL),

(37, 'Vulpix', 'Quando è giovane ha sei meravigliose code, che si moltiplicano durante la sua crescita', 'Erba alta', 25, 'fuoco', NULL),

(38, 'Ninetails', 'Dicono che viva un millennio. Ognuna delle sue code è dotata di un potere magico', 'Erba alta', 25, 'fuoco', NULL),

(52, 'Meowth', 'Ama raccogliere oggetti luccicanti. Quando è di buon umore mostra la sua collezione anche al suo Allenatore', 'Erba alta', 50, 'normale', NULL),

(95, 'Onix', 'Scava nel terreno assorbendo gli oggetti più duri per irrobustire il suo corpo', 'Caverna', 50, 'roccia', 'terra'),

(120, 'Staryu', 'Alla fine dell'estate, sui litorali è possibile osservare gruppi di Staryu che emettono luce a un ritmo regolare', 'Scava nel terreno assorbendo gli oggetti più duri per irrobustire il suo corpo', 'Spiaggia', NULL, 'acqua', NULL),

(121, 'Starmie', 'Quando scatena i suoi poteri psichici, l'organo centrale, detto nucleo, brilla di sette colori diversi', 'Spiaggia', NULL, 'acqua', 'psico'),

(133, 'Eevee', 'Ha la capacità di alterare la propria struttura corporea per adattarsi all'ambiente circostante', 'Erba alta', 87.5, 'normale', NULL),

(134, 'Vaporeon', 'Se le pinne di Vaporeon iniziano a vibrare, vuol dire che pioverà nel giro di poche ore', 'Erba alta', 87.5, 'acqua', NULL),

(135, 'Jolteon', 'Se si arrabbia o è sorpreso, tutti i peli del corpo gli si rizzano trasformandosi in aghi che infilzano il nemico', 'Erba alta', 87.5, 'elettro', NULL),

(136, 'Flareon', 'Quando immagazzina abbastanza calore, la sua temperatura corporea può salire fino a 900 °C', 'Erba alta', 87.5, 'fuoco', NULL),

(150, 'Mewtwo', 'Il suo DNA è quasi uguale a quello di Mew. Ciò nonostante, sono agli antipodi per dimensioni e carattere', 'Evento', NULL, 'psico', NULL),

(151, 'Mew', 'Osservando al microscopio la pelle di Mew si può constatare che è ricoperta da una fitta peluria, corta e fine', 'Evento', NULL, 'psico', NULL),

(152, 'Chikorita', 'Nella lotta, Chikorita sventola la sua foglia in modo da tenere a bada il nemico. Tuttavia, la foglia emana anche un aroma dolce, che calma il Pokémon

avversario, creando un''atmosfera gradevole e rilassata tutt''attorno', 'Evento', 87.5, 'erba', NULL),

(153, 'Bayleef', 'Il collo di Bayleef è agghindato da foglie, dentro cui è presente il piccolo germoglio di un albero. L''aroma emanato da questo germoglio ha un effetto energizzante sull''uomo', 'Evento', 87.5, 'erba', NULL),

(154, 'Meganium', 'L''aroma del fiore di Meganium placa le emozioni. Nella lotta questo Pokémon emana un profumo con effetto calmante per smorzare l''aggressività del nemico', 'Evento', 87.5, 'erba', NULL),

(155, 'Cyndaquil', 'Cyndaquil si protegge grazie alle fiamme che gli ardono sul dorso. Le fiamme divampano impetuose se il Pokémon è adirato. Tuttavia, quando è stanco diventano fiammelle innocue dalla combustione incompleta', 'Evento', 87.5, 'fuoco', NULL),

(156, 'Quilava', 'Quilava tiene a bada il nemico grazie all''intensità delle sue fiamme e a getti di aria rovente. Questo Pokémon sfrutta la sua estrema destrezza per evitare gli attacchi anche mentre ustiona il nemico', 'Evento', 87.5, 'fuoco', NULL),

(157, 'Typhlosion', 'Typhlosion si nasconde avvolto da una lucente nube di calore creata dalle sue fiamme roventi. Questo Pokémon crea esplosioni spettacolari che riducono in cenere ogni cosa', 'Evento', 87.5, 'fuoco', NULL),

(158, 'Totodile', 'Nonostante il suo corpicino minuto, le mascelle di Totodile sono molto potenti. Sebbene creda solo di giocare, il suo morso è così forte e pericoloso da causare serie ferite', 'Evento', 87.5, 'acqua', NULL),

(159, 'Croconaw', 'Una volta azzannato il nemico, Croconaw non molla la presa facilmente. Le zanne a punta ricurva all''indietro, a mo'' di amo da pesca, rendono impossibile estrarle quando sono conficcate in profondità', 'Evento', 87.5, 'acqua', NULL),

(160, 'Feraligatr', 'Feraligatr spaventa il nemico spalancando la sua bocca enorme. Nella lotta pesta pesantemente il terreno con le sue possenti zampe posteriori per scagliarsi contro il nemico a un''incredibile velocità', 'Evento', 87.5, 'acqua', NULL),

(196, 'Espeon', 'Analizza le correnti d'aria per predire le condizioni atmosferiche o la prossima mossa del nemico', 'Erba alta', 87.5, 'psico', NULL),

(197, 'Umbreon', 'Quando si arrabbia, spruzza dai pori un sudore velenoso mirando agli occhi del suo avversario', 'Erba alta', 87.5, 'buio', NULL),

(243, 'Raikou', 'Raikou rappresenta la velocità del fulmine. Il suo ruggito crea terrificanti onde d'urto nell'aria e scuote il suolo come se fosse percosso dalla furia di un fulmine durante il temporale', 'Erba alta', NULL, 'elettro', NULL),

(244, 'Entei', 'Entei rappresenta l'ardore del magma. Si narra che questo Pokémon sia nato dall'eruzione di un vulcano. Emette lingue di fuoco così violente da incenerire tutto ciò che lambiscono', 'Erba alta', NULL, 'fuoco', NULL),

(245, 'Suicune', 'Suicune rappresenta la purezza delle sorgenti d'acqua dolce. Corre con grazia immerso nella natura. Questo Pokémon ha la facoltà di depurare le acque di scarico', 'Erba alta', NULL, 'acqua', NULL),

(249, 'Lugia', 'Le ali di Lugia hanno un'enorme potenza distruttiva: un leggero battito d'ali è in grado di abbattere una normale abitazione. Perciò il Pokémon ha scelto di vivere lontano dall'uomo, nelle profondità abissali', 'Evento', NULL, 'psico', 'volante'),

(250, 'Ho-oh', 'Il piumaggio di Ho-Oh brilla in sette colori a seconda dell'angolazione dei raggi luminosi. Si dice che le sue piume portino felicità a chi le possiede. Si narra che viva ai piedi di un arcobaleno', 'Evento', NULL, 'fuoco', 'volante');

INSERT INTO evoluzione(da, a, livello, richieste) VALUES

(1, 2, 16, NULL),-- Bulbasaur -> Ivysaur -> Venusaur

(2, 3, 32, NULL),

(4, 5, 16, NULL),-- Charmander -> Charmeleon -> Charizard

(5, 6, 36, NULL),

(7, 8, 16, NULL),-- Squirtle -> Wartortle -> Blastoise

(8, 9, 36, NULL),

```

(25, 26, NULL, 'Use Thunder Stone'), -- Pikachu -> Raichu
(37, 38, NULL, 'Use Fire Stone'), -- Vulpix -> Ninetails
(133, 134, NULL, 'Use Water Stone'), -- Eevee -> Vaporeon
(133, 135, NULL, 'Use Thunder Stone'), -- Eevee -> Jolteon
(133, 136, NULL, 'Use Fire Stone'), -- Eevee -> Flareon
(152, 153, 16, NULL), -- Chikorita -> Bayleef -> Meganium
(153, 154, 32, NULL),
(155, 156, 16, NULL), -- Cyndaquil -> Quilava ->
Typhlosion
(156, 157, 32, NULL),
(158, 159, 16, NULL), -- Totodile -> Croconaw ->
Feraligatr
(159, 160, 32, NULL),
(133, 196, NULL, 'Have high friendship (day)'), -- Eevee
-> Espeon
(133, 197, NULL, 'Have high friendship (night)'), -- Eevee
-> Umbreon

```

```

INSERT INTO mosca(nome, tipo, is_tm, categoria, potenza)
VALUES

```

```

('Graffio', 'normale', false, 'Fisica', 40),
-----Gen I
('Frustata', 'erba', false, 'Fisica', 45),
('Azione', 'normale', false, 'Fisica', 40),
('Riduttore', 'normale', true, 'Fisica', 90),
('Sdoppiatore', 'normale', true, 'Fisica', 120),
('Colpocoda', 'normale', false, 'Stato', 0),
('Morso', 'buio', false, 'Fisica', 60),
('Ruggito', 'normale', false, 'Stato', 0),
('Braciore', 'fuoco', false, 'Speciale', 40),
('Lanciafiamme', 'fuoco', false, 'Speciale', 90),
('Pistolacqua', 'acqua', false, 'Speciale', 40),
('Idropompa', 'acqua', false, 'Speciale', 110),
('Parassiseme', 'erba', false, 'Stato', 0),
('Crescita', 'normale', false, 'Stato', 0),
('Foglielama', 'erba', false, 'Fisica', 55),
('Solarraggio', 'erba', true, 'Speciale', 120),

```

```

('Velenpolvere', 'veleno', false, 'Stato', 0),
('Sonnifero', 'erba', false, 'Stato', 0),
('Petalodanza', 'erba', false, 'Speciale', 120),
('Ira di Drago', 'drago', false, 'Speciale', 0),
('Ira', 'normale', true, 'Fisica', 20),
('Muro di Fumo', 'normale', false, 'Stato', 0),
('Ritirata', 'acqua', false, 'Stato', 0),
('Lacerazione', 'normale', false, 'Fisica', 70),
('Bolla', 'acqua', false, 'Speciale', 40),
('Capocciata', 'normale', false, 'Fisica', 130),
('Profumino', 'normale', false, 'Stato', 0),
-----Gen II
('Sintesi', 'erba', false, 'Stato', 0),
('Visotruce', 'normale', false, 'Stato', 0),
('Rapigiro', 'normale', false, 'Fisica', 50),
('Protezione', 'normale', false, 'Stato', 0),
('Pioggiadanza', 'acqua', false, 'Stato', 0);

INSERT INTO imparamossa(pokemon, mossa, partenza) VALUES
(1, 'Azione', true),
(1, 'Ruggito', false),
(1, 'Parassiseme', false),
(1, 'Frustata', false),
(1, 'Velenpolvere', false),
(1, 'Sonnifero', false),
(1, 'Foglielama', false),
(1, 'Profumino', false),
(1, 'Crescita', false),
(1, 'Sintesi', false),
(1, 'Solarraggio', false),
(2, 'Riduttore', false),
(2, 'Sdoppiatore', false),
(3, 'Petalodanza', false),
(4, 'Graffio', false),
(4, 'Ruggito', false),
(4, 'Bracieme', false),
(4, 'Muro di Fumo', false),
(4, 'Ira', false),

```

```

(4, 'Visotruce', false),
(5, 'Lanciafiamme', false),
(5, 'Lacerazione', false),
(6, 'Ira di Drago', false),
(7, 'Azione', true),
(7, 'Colpocoda', false),
(7, 'Bolla', false),
(7, 'Ritirata', false),
(7, 'Pistolacqua', false),
(8, 'Morso', false),
(8, 'Rapigiro', false),
(8, 'Protezione', false),
(9, 'Pioggiadanza', false),
(9, 'Capocciata', false),
(9, 'Idropompa', false);

```

```

INSERT INTO ZonaCatturaPokemonGo(stato, citta, tipo_zona,
pokemon) VALUES
('Italia', 'Modena', 'Urbano', 151), -- Mew
('Italia', 'Sozzigalli', 'Rurale', 37), -- Vulpix
('Italia', 'Rimini', 'Mare', 120); -- Staryu

```

```

INSERT INTO gioco(nome, data_rilascio, casa_sviluppo) VALUES
('Pokemon Red/Blue', '1996-02-27', 'Nintendo'),
('Pokemon Gold/Silver', '1999-11-21', 'Nintendo');

```

```

INSERT INTO pokemoningioco(pokemon, gioco, is_leggendario)
VALUES
(1, 'Pokemon Red/Blue', false),
(2, 'Pokemon Red/Blue', false),
(3, 'Pokemon Red/Blue', false),
(4, 'Pokemon Red/Blue', false),
(5, 'Pokemon Red/Blue', false),
(6, 'Pokemon Red/Blue', false),
(7, 'Pokemon Red/Blue', false),
(8, 'Pokemon Red/Blue', false),
(9, 'Pokemon Red/Blue', false),

```

(25, 'Pokemon Red/Blue', false),
(26, 'Pokemon Red/Blue', false),
(37, 'Pokemon Red/Blue', false),
(38, 'Pokemon Red/Blue', false),
(95, 'Pokemon Red/Blue', false),
(120, 'Pokemon Red/Blue', false),
(121, 'Pokemon Red/Blue', false),
(133, 'Pokemon Red/Blue', false),
(134, 'Pokemon Red/Blue', false),
(135, 'Pokemon Red/Blue', false),
(136, 'Pokemon Red/Blue', false),
(150, 'Pokemon Red/Blue', true),
(151, 'Pokemon Red/Blue', true),
(1, 'Pokemon Gold/Silver', false),
(2, 'Pokemon Gold/Silver', false),
(3, 'Pokemon Gold/Silver', false),
(4, 'Pokemon Gold/Silver', false),
(5, 'Pokemon Gold/Silver', false),
(6, 'Pokemon Gold/Silver', false),
(7, 'Pokemon Gold/Silver', false),
(8, 'Pokemon Gold/Silver', false),
(9, 'Pokemon Gold/Silver', false),
(25, 'Pokemon Gold/Silver', false),
(26, 'Pokemon Gold/Silver', false),
(37, 'Pokemon Gold/Silver', false),
(38, 'Pokemon Gold/Silver', false),
(95, 'Pokemon Gold/Silver', false),
(120, 'Pokemon Gold/Silver', false),
(121, 'Pokemon Gold/Silver', false),
(133, 'Pokemon Gold/Silver', false),
(134, 'Pokemon Gold/Silver', false),
(135, 'Pokemon Gold/Silver', false),
(136, 'Pokemon Gold/Silver', false),
(150, 'Pokemon Gold/Silver', false),
(151, 'Pokemon Gold/Silver', false),
(152, 'Pokemon Gold/Silver', false),
(153, 'Pokemon Gold/Silver', false),
(154, 'Pokemon Gold/Silver', false),

```

(155, 'Pokemon Gold/Silver', false),
(156, 'Pokemon Gold/Silver', false),
(157, 'Pokemon Gold/Silver', false),
(158, 'Pokemon Gold/Silver', false),
(159, 'Pokemon Gold/Silver', false),
(160, 'Pokemon Gold/Silver', false),
(196, 'Pokemon Gold/Silver', false),
(197, 'Pokemon Gold/Silver', false),
(243, 'Pokemon Gold/Silver', false),
(244, 'Pokemon Gold/Silver', true),
(245, 'Pokemon Gold/Silver', true),
(249, 'Pokemon Gold/Silver', true),
(250, 'Pokemon Gold/Silver', true);

```

```

INSERT INTO dispositivoconsole (nome, data_rilascio,
portatile) VALUES
('Game Boy', '1989-04-21', true),
('Game Boy Color', '1998-08-23', true);

```

```

INSERT INTO pubblicazioneconsole(gioco, dispositivo) VALUES
('Pokemon Red/Blue', 'Game Boy'),
('Pokemon Gold/Silver', 'Game Boy Color');

```

```

INSERT INTO personaggio (nome, cognome, rappresenta_pokemon)
VALUES
('Ash', 'Ketchum', NULL),
('Misty', 'Williams', NULL),
('Brock', 'Harrison', NULL),
('Rosso', '', NULL),
('Jesse', '', NULL),
('James', '', NULL),
('Meowth', '', 52),
('Pikachu', '', 25);

```

```

INSERT INTO pokemonpossesso (personaggio_nome,
personaggio_cognome, pokemon) VALUES
('Ash', 'Ketchum', 25), -- Pikachu
('Ash', 'Ketchum', 6), -- Charizard

```

```

('Misty', 'Williams', 120), -- Staryu
('Brock', 'Harrison', 95), -- Onix
('Brock', 'Harrison', 37); -- Vulpix

```

```

INSERT INTO multimedia (nome, anno, casa_produzione, creatore,
tipo_multimediale) VALUES

```

```

('Pokémon il film - Mewtwo colpisce ancora', 1998,
'Oriental Light And Magic', 'Kunihiko Yuyama', 'Film anime'),
('Pokémon 2 - La forza di uno', 1999, 'Oriental Light And
Magic', 'Kunihiko Yuyama', 'Film anime'),
('Pocket Monsters Sekiei League', 1997, 'Oriental Light
And Magic', 'Takeshi Shudō', 'Serie anime'),
('Pokémon Adventures', 2002, 'Shogakukan', 'Hidenori
Kusaka', 'Manga'),
('Pokémon 3 - L''incantesimo degli Unown', 2000,
'Oriental Light And Magic', 'Kunihiko Yuyama', 'Film anime');

```

```

INSERT INTO comparein (pokemon, multimedia_nome,
multimedia_anno) VALUES

```

```

(150, 'Pokémon il film - Mewtwo colpisce ancora', 1998),
-- Mewtwo
(151, 'Pokémon il film - Mewtwo colpisce ancora', 1998),
-- Mew
(25, 'Pokémon il film - Mewtwo colpisce ancora', 1998),
-- Pikachu
(37, 'Pokémon il film - Mewtwo colpisce ancora', 1998),
-- Vulpix
(120, 'Pokémon il film - Mewtwo colpisce ancora', 1998),
-- Staryu
(52, 'Pokémon il film - Mewtwo colpisce ancora', 1998),
-- Meowth
(25, 'Pokémon 2 - La forza di uno', 1999), -- Pikachu
(1, 'Pokémon 2 - La forza di uno', 1999), -- Bulbasaur
(6, 'Pokémon 2 - La forza di uno', 1999), -- Charizard
(7, 'Pokémon 2 - La forza di uno', 1999), -- Squirtle
(120, 'Pokémon 2 - La forza di uno', 1999), -- Staryu
(1, 'Pocket Monsters Sekiei League', 1997), -- Bulbasaur
(4, 'Pocket Monsters Sekiei League', 1997), -- Charmander

```



```

(7, 'Pocket Monsters Sekiei League', 1997), -- Squirtle
(25, 'Pocket Monsters Sekiei League', 1997), -- Pikachu
(52, 'Pocket Monsters Sekiei League', 1997), -- Meowth
(25, 'Pokémon Adventures', 2002), -- Pikachu
(150, 'Pokémon Adventures', 2002), -- Mewtwo
(4, 'Pokémon Adventures', 2002); -- Charmander

```

```

INSERT INTO comparsa (personaggio_nome, personaggio_cognome,
multimedia_nome, multimedia_anno) VALUES
('Ash', 'Ketchum', 'Pokémon il film - Mewtwo
colpisce ancora', 1998),
('Misty', 'Williams', 'Pokémon il film - Mewtwo
colpisce ancora', 1998),
('Brock', 'Harrison', 'Pokémon il film - Mewtwo
colpisce ancora', 1998),
('Pikachu', '', 'Pokémon il film - Mewtwo colpisce
ancora', 1998),
('Ash', 'Ketchum', 'Pokémon 2 - La forza di uno',
1999),
('Misty', 'Williams', 'Pokémon 2 - La forza di uno',
1999),
('Brock', 'Harrison', 'Pokémon 2 - La forza di uno',
1999),
('Pikachu', '', 'Pokémon 2 - La forza di uno', 1999),
('Ash', 'Ketchum', 'Pokémon 3 - L''incantesimo
degli Unown', 2000),
('Misty', 'Williams', 'Pokémon 3 - L''incantesimo degli
Unown', 2000),
('Brock', 'Harrison', 'Pokémon 3 - L''incantesimo degli
Unown', 2000),
('Pikachu', '', 'Pokémon 3 - L''incantesimo degli
Unown', 2000),
('Ash', 'Ketchum', 'Pocket Monsters Sekiei League',
1997),
('Misty', 'Williams', 'Pocket Monsters Sekiei League',
1997),
('Brock', 'Harrison', 'Pocket Monsters Sekiei League',
1997),

```

```
( 'Jesse', '', 'Pocket Monsters Sekiei League', 1997),
( 'James', '', 'Pocket Monsters Sekiei League', 1997),
( 'Meowth', '', 'Pocket Monsters Sekiei League', 1997),
( 'Pikachu', '', 'Pocket Monsters Sekiei League', 1997),
( 'Rosso', '', 'Pokémon Adventures', 2002);
```

```
INSERT INTO doppiatore (cod_fis, nome, cognome, data_nascita)
VALUES
```

```
( 'GRBDVD68P24C722H', 'Davide', 'Garbolino',
'1968-09-24'),
( 'KRPLSN63D70F205K', 'Alessandra', 'Karpoff',
'1963-04-30'),
( 'CRRNCL63D30E463K', 'Nicola', 'Carrassi', '1971-08-01'),
( 'MTSRCI68S70F257Y', 'Rica', 'Matsumoto', '1968-11-30'),
( 'BTTLCU67L01F2050', 'Luca', 'Bottale', '1967-07-01'),
( 'NTCSRH87P60A944R', 'Sarah', 'Natochenny',
'1987-09-20'),
( 'BLDPTR55P21F205G', 'Pietro', 'Ubaldi', '1955-09-21'),
( 'CLVGPP62T09L781I', 'Giuseppe', 'Calvetti',
'1962-12-09'),
( 'TNOKIU65M18L424U', 'Ikue', 'Otani', '1965-08-18');
```

```
INSERT INTO doppiaggio (personaggio_nome, personaggio_cognome,
multimedia_nome, multimedia_anno, lingua, doppiatore) VALUES
```

```
( 'Ash', 'Ketchum', 'Pokémon il film - Mewtwo colpisce
ancora', 1998, 'Italiano', 'GRBDVD68P24C722H'),
( 'Ash', 'Ketchum', 'Pocket Monsters Sekiei League',
1997, 'Italiano', 'GRBDVD68P24C722H'),
( 'Ash', 'Ketchum', 'Pokémon 2 - La forza di uno', 1999,
'Italiano', 'GRBDVD68P24C722H'),
( 'Ash', 'Ketchum', 'Pokémon 3 - L''incantesimo degli
Unown', 2000, 'Italiano', 'GRBDVD68P24C722H'),
( 'Ash', 'Ketchum', 'Pokémon il film - Mewtwo colpisce
ancora', 1998, 'Giapponese', 'MTSRCI68S70F257Y'),
( 'Ash', 'Ketchum', 'Pokémon il film - Mewtwo colpisce
ancora', 1998, 'Inglese', 'NTCSRH87P60A944R'),
( 'Misty', 'Williams', 'Pokémon il film - Mewtwo colpisce
ancora', 1998, 'Italiano', 'KRPLSN63D70F205K'),
```

('Misty', 'Williams', 'Pocket Monsters Sekiei League',
 1997, 'Italiano', 'KRPLSN63D70F205K'),
 ('Misty', 'Williams', 'Pokémon 2 - La forza di uno',
 1999, 'Italiano', 'KRPLSN63D70F205K'),
 ('Misty', 'Williams', 'Pokémon 3 - L''incantesimo degli
 Unown', 2000, 'Italiano', 'KRPLSN63D70F205K'),
 ('Brock', 'Harrison', 'Pokémon il film - Mewtwo colpisce
 ancora', 1998, 'Italiano', 'CRRNCL63D30E463K'),
 ('Brock', 'Harrison', 'Pocket Monsters Sekiei League',
 1997, 'Italiano', 'CRRNCL63D30E463K'),
 ('Brock', 'Harrison', 'Pocket Monsters Sekiei League',
 1997, 'Italiano', 'BTTLCU67L01F2050'),
 ('Brock', 'Harrison', 'Pokémon 3 - L''incantesimo degli
 Unown', 2000, 'Italiano', 'BTTLCU67L01F2050'),
 ('Meowth', '', 'Pocket Monsters Sekiei League', 1997,
 'Italiano', 'CLVGPP62T09L781I'),
 ('Meowth', '', 'Pocket Monsters Sekiei League', 1997,
 'Italiano', 'BLDPTR55P21F205G'),
 ('Pikachu', '', 'Pokémon il film - Mewtwo colpisce
 ancora', 1998, 'Giapponese', 'TNOKIU65M18L424U'),
 ('Pikachu', '', 'Pokémon il film - Mewtwo colpisce
 ancora', 1998, 'Italiano', 'TNOKIU65M18L424U'),
 ('Pikachu', '', 'Pokémon il film - Mewtwo colpisce
 ancora', 1998, 'Inglese', 'TNOKIU65M18L424U'),
 ('Pikachu', '', 'Pocket Monsters Sekiei League',
 1997, 'Giapponese', 'TNOKIU65M18L424U'),
 ('Pikachu', '', 'Pocket Monsters Sekiei League',
 1997, 'Italiano', 'TNOKIU65M18L424U'),
 ('Pikachu', '', 'Pocket Monsters Sekiei League',
 1997, 'Inglese', 'TNOKIU65M18L424U');

Studio di alcune query SQL

1) Attacchi più forti della media

```
select * from mossa
where potenza > (select avg(potenza) from mossa);
```

2) Trovare il personaggio doppiato da doppiatori diversi nella stessa lingua e nello stesso multimedia

```
select
dopp.*, doppiatore.nome, doppiatore.cognome, doppiatore.data_nascita
from (select distinct d1.* from doppiaggio as d1, doppiaggio as d2
where d1.personaggio_nome=d2.personaggio_nome and
d1.personaggio_cognome=d2.personaggio_cognome and
d1.multimedia_nome=d2.multimedia_nome and
d1.multimedia_anno=d2.multimedia_anno and
d1.lingua=d2.lingua and d1.doppiatore<>d2.doppiatore) as dopp
join doppiatore on dopp.doppiatore=doppiatore.cod_fis;
```

3) Seleziona per ogni personaggio il Pokémon che rappresenta (se esiste), il suo id, il numero di Pokémon che possiede e il numero di comparse

```
select q.*, p.nome,
(select count(*) from pokemonpossesso
where personaggio_nome=q.nome and
personaggio_cognome=q.cognome) as pcnt,
(select count(*) from comparsa
where personaggio_nome=q.nome and
personaggio_cognome=q.cognome) as pcmp
from personaggio as q
left join pokemon as p on q.rappresenta_pokemon=p.id;
```

4) Raggruppare i Pokémon in base alla percentuale del sesso con il quale si possono trovare

```
select count(*) as nPokemon,pokemon.percentuale_sesso from
pokemon
group by pokemon.percentuale_sesso;
```

5) Individuare i Pokémon con più di un evoluzione

```
select da, count(*) as nEvoluzioni from evoluzione
group by da
having count(*) > 1;
```

6) Ordinare le mosse in base alla potenza

```
select * from mossa
order by potenza desc;
```

7) Mostrare il doppiatore che ha doppiato un personaggio in tutte le lingue disponibili

```
select * from doppiatore
where not exists ( select lingua from doppiaggio as d1
    where not exists (
        select * from doppiaggio as d2
        where d1.lingua=d2.lingua and
doppiatore.cod_fis=d2.doppiatore
    )
);
```

8) Trovare il tipo predefinito di Pokémon più adatto per affrontarne un altro scelto

```
select confrontotipo.* from (select da, max(efficacia) as eff
from confrontotipo
group by da) as parte
join confrontotipo on confrontotipo.efficacia = parte.eff and
confrontotipo.da=parte.da;
```

9) Seleziona dato un pokemon i suoi dati (nome, zona_cattura, percentuale_sesso, tipo1, tipo2), id e nome del Pokémon padre (da cui

è evoluto), numero di sotto-evoluzioni e numero di personaggi che lo rappresentano

```
select p1.*,p2.id,p2.nome,  
(select count(*) from evoluzione e1 where e1.da=p1.id) as  
evolcnt,  
(select count(*) from personaggio where  
personaggio.rappresenta_pokemon=p1.id) as rapprcnt  
from pokemon p1  
left join evoluzione as e ON p1.id = e.a  
left join pokemon as p2 ON p2.id = e.da  
order by p1.id;
```

10) Dato il nome del gioco elencare i Pokémon leggendari presenti al suo interno

```
select PG.gioco, P.* from pokemoningioco as PG  
join pokemon as P ON P.id=PG.pokemon  
where PG.is_leggendario=true  
order by PG.gioco,P.id;
```

Esempio di applicazione

L'applicazione d'esempio è "Pokexp", una semplice applicazione in Python che offre un'interfaccia testuale (TUI) per l'esplorazione del database. L'unica dipendenza richiesta è "psycpg2" (dal pacchetto "psycpg2-binary") per la connessione a PostgreSQL e un ambiente funzionante di Python 3.

L'applicazione supporta sia le query studiate sia l'esplorazione relazionale libera (passare da un'entità a quelle a lei collegate).

Inoltre è possibile l'aggiunta e l'eliminazione guidate di tuple selezionate dall'utente.

Per i dettagli sulla modalità corretta di utilizzo del software consultare il file "README.pdf" presente nella cartella del programma.

```
20: Flareon
21: Mewtwo
22: Mew
23: Chikorita
24: Bayleef
25: Meganium
26: Cyndaquil
27: Quilava
28: Typhlosion
29: Totodile
Page: [< 2 >]
Input (b=back)>■
```

```
Charmeleon
Ha un'indole feroce. Usa la coda fiammeggiante
come una frusta e lacerava l'avversario con gli
artigli affilati.
National Pokedex: 5
Trovabile in: Evento
Sesso: 87.5% maschio, 12.5% femmina
Tipo: fuoco
Evoluto da: Charmander
0: Mosse
1: Cerca in Pokemon GO
2: Charmander
3: Evoluzioni (1)
4: Comparsa multimediali
5: Appartenenza personaggi
6: Indietro
Input (b=back)>■
```

```
da      |contro |efficacia
erba    |roccia |2
ghiaccio|erba   |2
ghiaccio|terra  |2
ghiaccio|volante|2
lotta  |normale|2
lotta  |erba   |2
lotta  |roccia |2
lotta  |buio   |2
veleno  |erba   |2
terra   |fuoco  |2
Page: [< 1 >]
Input>■
```

```
0. nome=Pokemon Giallo
1. num_pokemon=0
2. data_rilascio=1998-09-12
3. casa_sviluppo=Game Freak
[g=go, b=back]
Insert>■
```