COMP3311 23T1 Database Systems

# Assignment 2
## The Pokémon Database

Last updated: **Thursday 30th March 11:14pm**
Most recent changes are shown in red ... older changes are shown in brown.
[Assignment Spec] **[Database Design]** [Examples] [Testing] [Submitting] [Fixes+Updates]

## Introduction

This document contains a description of the Pokémon database. We give both an ER model and an SQL schema.

The database contains a large number of tables and relationships. We summarise what each table represents and the kind of information it holds here. More information is given in the downloadable schema.

**Pokemon**
This table describes general aspects of each Pokémon, including: its unique ID (a combination of a Pokédex number and a variation number), its name, its species, its growth rate, its basic properties (hit points, speed, etc.), and so on.

**Games**
This table indicates which region a particular game occurs in.
The game ID is also used as part of a Pokédex, and an Encounter.

**Types**
Each Pokémon has at least one type (e.g. fire, water, ghost, flying).
Some Pokémon may have two types.
All Moves also have a type.

**Abilities**
Pokémon have a large range of possible abilities (e.g. flame body, gooey, iron fist, neuroforce).
A description of each ability is contained in the `effect` column of this table.
Each Ability a Pokémon knows is given in the `Knowable_Abilities` table.

**Moves**
Pokémon aso have a large range of possible moves they can make (e.g. blizzard, block, bounce, etc.).
Each move has an associated category, power and accuracy.
Pokémon can potentially learn more move during a game.
Each move a Pokémon can learn given in the `Learnable_Moves` table.

**Evolutions**
Some Pokémon can change form, and this table describes the starting form and final form.
What conditions are needed before this change can occur, is given in the `Evolution_Requirements` table.

**Requirements**
Some changes (evolutions, encounters, learnable moves) require certain pre-conditions before they can occur.
The `Requirements` table gives a list of possible pre-conditions which can be applied to the various table associated with "changes".

**Encounters**
Encounters describe where you might find a Pokémon under certain circumstances, how likely is the encounter, and at what level the Pokémon may be.

**Egg Groups**

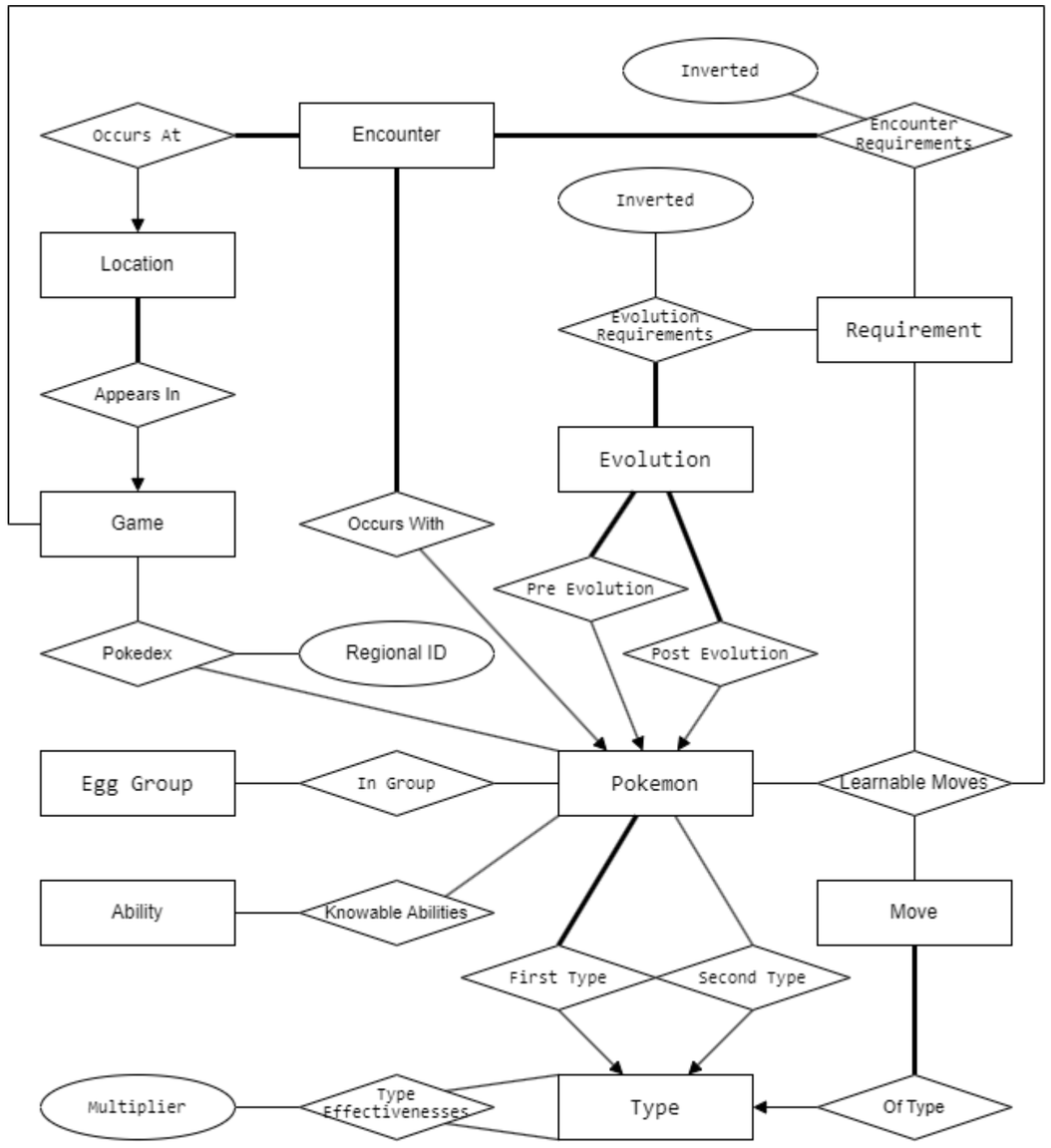
Pokémon can breed, but only with other Pokémon in the same Egg Group.

## ER Models

The following ER diagrams show the major components of the Pokémon database:

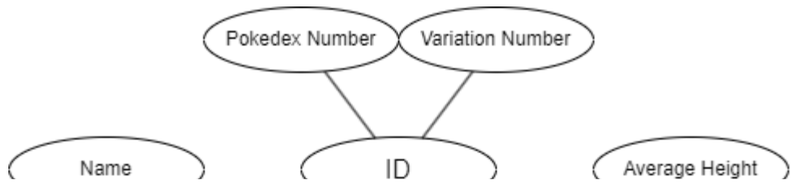
ER diagram of Pokémon relations

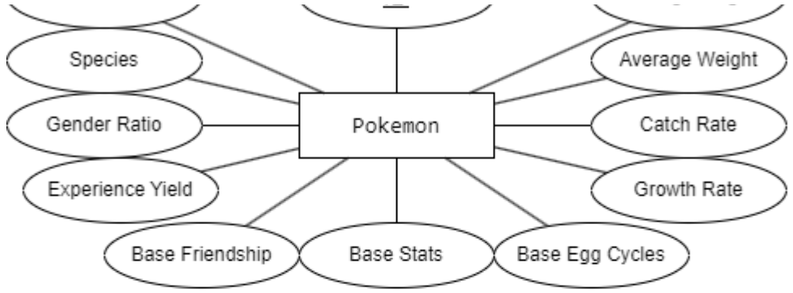
Attributes have been removed to make the relations more readable

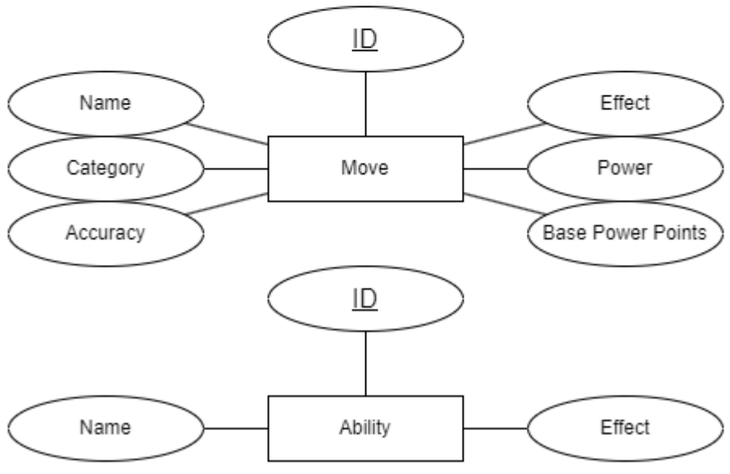


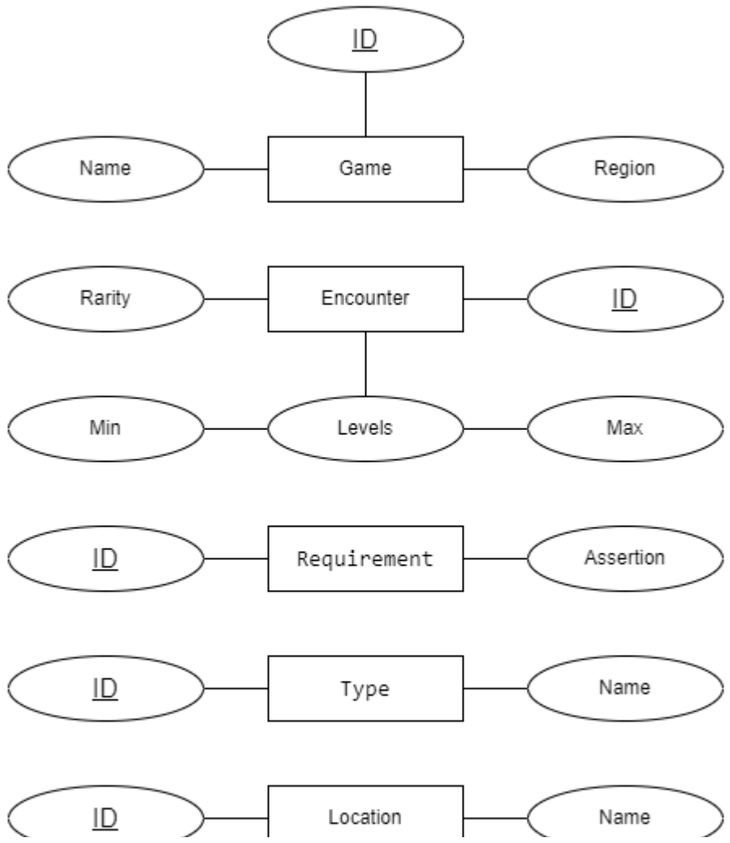

ER diagram of Pokémon attributes

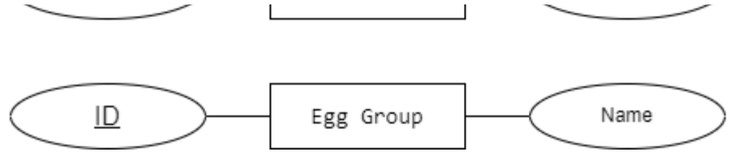
Relations have been removed to make the attributes more readable




Species
Gender Ratio
Experience Yield
Base Friendship
Pokemon
Base Stats
Base Egg Cycles
Average Weight
Catch Rate
Growth Rate


ID
Name
Category
Accuracy
Move
Effect
Power
Base Power Points
ID
Name
Ability
Effect


ID
Name
Game
Region
Rarity
Encounter
ID
Min
Levels
Max
ID
Requirement
Assertion
ID
Type
Name
ID
Location
Name




## SQL Schema

The following is the SQL schema used in building the Pokémon database:

```
-- COMP3311
-- 23T1
-- Assignment 2
-- Pokemon Database

-- Schema By: Dylan Brotherston <d.brotherston@unsw.edu.au>
-- Version 1.0
-- 2023-03-25

CREATE DOMAIN Byte AS
    SMALLINT
    CHECK (
        VALUE >= 0
        AND
        VALUE <= 255
    )
;

CREATE DOMAIN Statistic AS
    Byte
    CHECK (
        VALUE >= 1
    )
;

CREATE DOMAIN Percentage AS
    INTEGER
    DEFAULT 100
    CHECK (
        VALUE >= 0
    )
;

CREATE DOMAIN Ratio AS
    Percentage
    DEFAULT 50
    CHECK (
        VALUE <= 100
    )
;

CREATE DOMAIN Probability AS
    Percentage
    DEFAULT 0
    CHECK (
```

```
        VALUE <= 100
    )
;

CREATE DOMAIN Meters AS
    REAL
    CHECK (
        VALUE >= 0
    )
;

CREATE DOMAIN Kilograms AS
    REAL
    CHECK (
        VALUE >= 0
    )
;

CREATE TYPE _Pokemon_ID AS (
    Pokedex_Number   INTEGER,
    Variation_Number INTEGER
);

CREATE DOMAIN Pokemon_ID AS
    _Pokemon_ID
    CHECK (
        (VALUE).Pokedex_Number   IS NOT NULL
        AND
        (VALUE).Variation_Number IS NOT NULL
    )
;

CREATE TYPE _Stats AS (
    Hit_Points      Statistic,
    Attack          Statistic,
    Defense         Statistic,
    Special_Attack  Statistic,
    Special_Defense Statistic,
    Speed           Statistic
);

CREATE DOMAIN Stats AS
    _Stats
    CHECK (
        (VALUE).Hit_Points      IS NOT NULL
        AND
        (VALUE).Attack          IS NOT NULL
        AND
        (VALUE).Defense         IS NOT NULL
        AND
        (VALUE).Special_Attack  IS NOT NULL
        AND
        (VALUE).Special_Defense IS NOT NULL
        AND
        (VALUE).Speed           IS NOT NULL
```

```
    )
;

CREATE TYPE _Range AS (
    MIN INTEGER,
    MAX INTEGER
);

CREATE DOMAIN Open_Range AS
    _Range
    CHECK (
        (VALUE).Min <= (VALUE).Max
        AND
        (
            (VALUE).Min IS NOT NULL
            OR
            (VALUE).Max IS NOT NULL
        )
    )
;

CREATE DOMAIN Closed_Range AS
    Open_Range
    CHECK (
        (VALUE).Min IS NOT NULL
        AND
        (VALUE).Max IS NOT NULL
    )
;

CREATE TYPE Growth_Rates AS ENUM (
    'Erratic',
    'Fast',
    'Medium Fast',
    'Medium Slow',
    'Slow',
    'Fluctuating'
);

CREATE TYPE Move_Categories AS ENUM (
    'Physical',
    'Special',
    'Status'
);

CREATE TYPE Regions AS ENUM (
    'Kanto',
    'Johto',
    'Hoenn',
    'Sinnoh',
    'Unova',
    'Kalos',
    'Alola',
    'Galar',
    'Hisui',
```

```
    'Paldea'
);

CREATE TABLE Types (
    ID   SERIAL          PRIMARY KEY,
    Name Text   NOT NULL UNIQUE
);

CREATE TABLE Type_Effectiveness (
    Attacking  INTEGER            REFERENCES Types (ID),
    Defending  INTEGER            REFERENCES Types (ID),
    Multiplier Percentage NOT NULL,
    PRIMARY KEY (Attacking, Defending)
);

CREATE TABLE Requirements (
    ID        SERIAL          PRIMARY KEY,
    Assertion Text   NOT NULL UNIQUE
);

CREATE TABLE Pokemon (
    ID              Pokemon_ID           PRIMARY KEY,
    Name            Text         NOT NULL UNIQUE,
    Species         Text         NOT NULL,
    First_Type      INTEGER      NOT NULL REFERENCES Types (ID),
    Second_Type     INTEGER               REFERENCES Types (ID),
    Average_Height  Meters       NOT NULL,
    Average_Weight  Kilograms    NOT NULL,
    Catch_Rate      Statistic    NOT NULL,
    Growth_Rate     Growth_Rates NOT NULL,
    Experience_Yield INTEGER     NOT NULL,
    Gender_Ratio    Ratio,
    Base_Stats      Stats        NOT NULL,
    Base_Friendship Byte         NOT NULL,
    Base_Egg_Cycles  INTEGER     NOT NULL
);

CREATE TABLE Egg_Groups (
    ID   SERIAL          PRIMARY KEY,
    Name Text   NOT NULL UNIQUE
);

CREATE TABLE In_Group (
    Pokemon   Pokemon_ID REFERENCES Pokemon (ID),
    Egg_Group INTEGER    REFERENCES Egg_Groups (ID),
    PRIMARY KEY (Pokemon, Egg_Group)
);

CREATE TABLE Evolutions (
    ID             SERIAL             PRIMARY KEY,
    Pre_Evolution  Pokemon_ID NOT NULL REFERENCES Pokemon (ID),
    Post_Evolution Pokemon_ID NOT NULL REFERENCES Pokemon (ID)
);

CREATE TABLE Evolution_Requirements (
```

```
211.     Evolution   INTEGER          REFERENCES Evolutions (ID),
212.     Requirement INTEGER          REFERENCES Requirements (ID),
213.     Inverted    BOOLEAN NOT NULL DEFAULT FALSE,
214.     PRIMARY KEY (Evolution, Requirement)
215. );
216.
217. CREATE TABLE Games (
218.     ID     SERIAL           PRIMARY KEY,
219.     Name   Text    NOT NULL UNIQUE,
220.     Region Regions NOT NULL
221. );
222.
223. CREATE TABLE Locations (
224.     ID         SERIAL           PRIMARY KEY,
225.     Name       Text    NOT NULL,
226.     Appears_In INTEGER NOT NULL REFERENCES Games (ID)
227. );
228.
229. CREATE TABLE Pokedex (
230.     National_ID Pokemon_ID          REFERENCES Pokemon (ID),
231.     Game        INTEGER              REFERENCES Games (ID),
232.     Regional_ID INTEGER    NOT NULL,
233.     PRIMARY KEY (National_ID, Game)
234. );
235.
236. CREATE TABLE Encounters (
237.     ID          SERIAL                PRIMARY KEY,
238.     Occurs_With Pokemon_ID   NOT NULL REFERENCES Pokemon (ID),
239.     Occurs_At   INTEGER       NOT NULL REFERENCES Locations (ID),
240.     Rarity      Probability  NOT NULL,
241.     Levels      Closed_Range NOT NULL
242. );
243.
244. CREATE TABLE Encounter_Requirements (
245.     Encounter   INTEGER          REFERENCES Encounters (ID),
246.     Requirement INTEGER          REFERENCES Requirements (ID),
247.     Inverted    BOOLEAN NOT NULL DEFAULT FALSE,
248.     PRIMARY KEY (Encounter, Requirement)
249. );
250.
251.
252. CREATE TABLE Moves (
253.     ID               SERIAL                  PRIMARY KEY,
254.     Name             Text            NOT NULL UNIQUE,
255.     Effect           Text,
256.     Of_Type          INTEGER         NOT NULL REFERENCES Types (ID),
257.     Category         Move_Categories,
258.     POWER             INTEGER,
259.     Accuracy         INTEGER,
260.     Base_Power_Points INTEGER
261. );
262.
263. CREATE TABLE Learnable_Moves (
264.     Learnt_By   Pokemon_ID NOT NULL REFERENCES Pokemon (ID),
265.     Learnt_In   INTEGER    NOT NULL REFERENCES Games (ID),
```

```
    Learnt_When INTEGER    NOT NULL REFERENCES Requirements (ID),
    Learns      INTEGER    NOT NULL REFERENCES Moves (ID),
    PRIMARY KEY (Learnt_By, Learnt_In, Learnt_When, Learns)
);

CREATE TABLE Abilities (
    ID     SERIAL          PRIMARY KEY,
    Name   Text   NOT NULL UNIQUE,
    Effect Text   NOT NULL
);

CREATE TABLE Knowable_Abilities (
    Known_By Pokemon_ID          REFERENCES Pokemon (ID),
    Knows    INTEGER             REFERENCES Abilities (ID),
    Hidden  BOOLEAN     NOT NULL,
    PRIMARY KEY (Known_By, Knows)
);

```

In an attempt to make the schema more concise, comments have been removed from this schema.