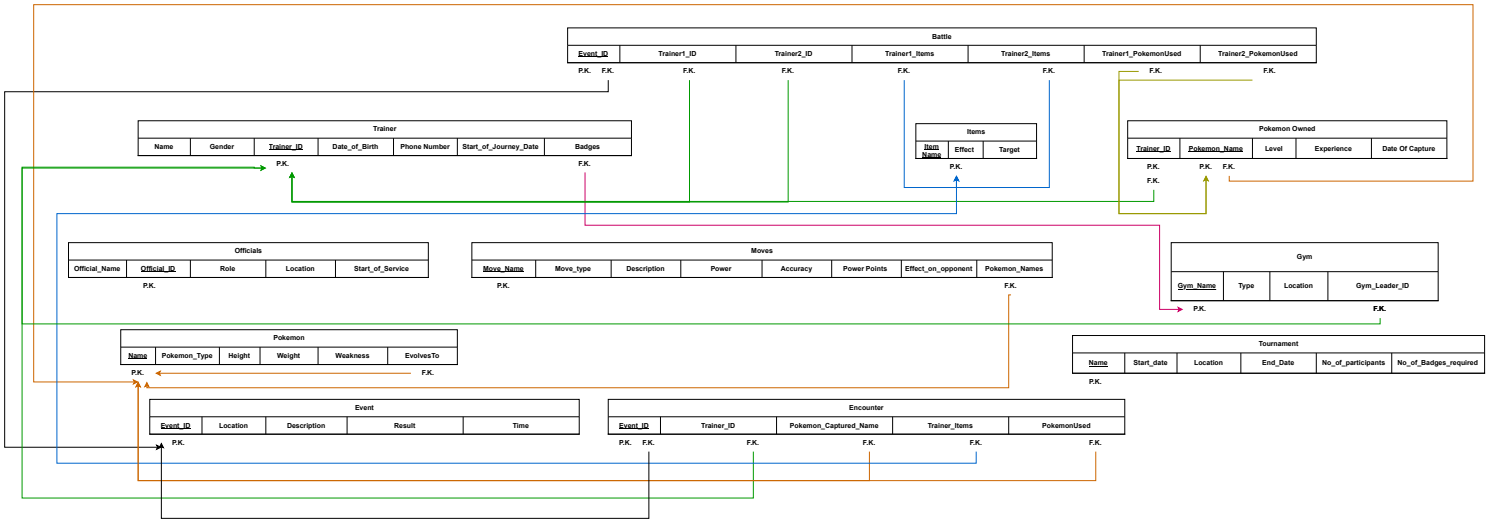
DnA Project Phase 3 Team 30

Relational Model:

We need to add additional attributes and tables to properly store and display the relationships.

- We have added a badges attribute to Trainers to encode the HasBadges relationship, a multi-valued attribute containing the names of all gyms the trainer has badges of.
- We have added a gym leader ID attribute to the Gym table identifying the trainer ID of the gym leader, encoding the IsGymLeader relationship.
- We have added a Pokemon names attribute to Moves table, a multi-valued attribute identifying all the pokemons with that particular move, encoding the HasMoves relationship.
- We have added a EvolvesTo attribute to the Pokemon table, a multi-valued attribute detailing its next possible evolutions, encoding the EvolvesTo relationship.
- We have added three tables Event, Encounter and Battle, where Event is the superclass and Encounter and Battle are its subclasses. The Event Table contains its attributes as defined in the diagram.
- The Encounter Table has an Event ID attribute as a foreign key to reference the event. The rest of the attributes encode the HasEncounter relationship, with TrainerItems and PokemonUsed being multi-value. The PokemonUsed relationship is also encoded here.
- The Battle Table has an EventID attribute as a foreign key to reference the event. The rest of the attributes encode the HasBattle relationship, with Traienr1_Items, Trainer2_Items, Trainer1_PokemonUsed and Trainer2_pokemonUsed being multi-valued. The PokemonUsed relationship is also encoded here.
- We have added a Pokemon Owned table, detailing all the pokemons a particular trainer has, and their level, experience and date of capture as detailed in the HasPokemon relationship with its attributes.

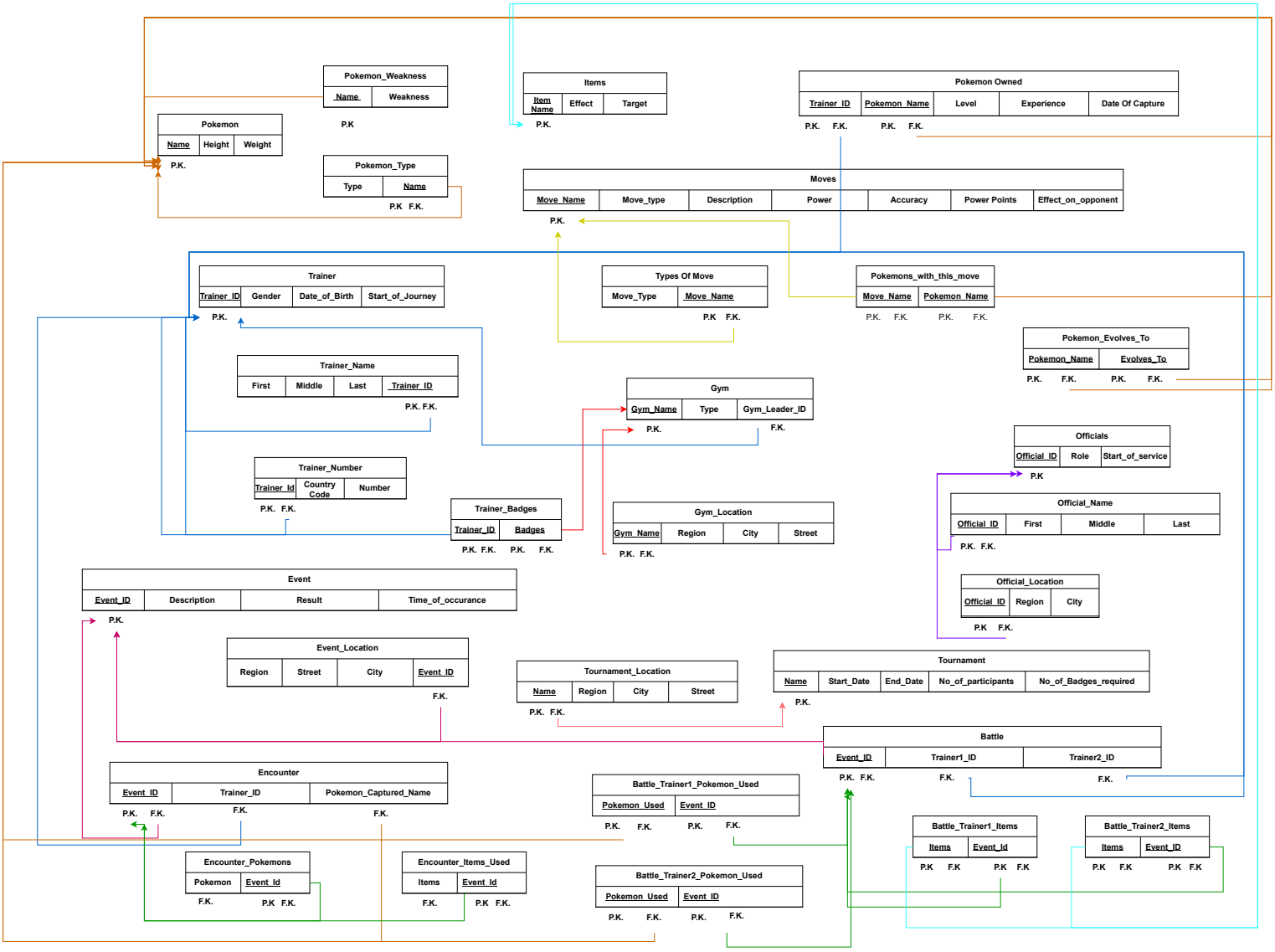

Battle
Event_ID
Trainer1_ID
Trainer2_ID
Trainer1_Items
Trainer2_Items
Trainer1_PokemonUsed
Trainer2_PokemonUsed
P.K.
F.K.
Trainer
Name
Gender
Trainer_ID
Date_of_Birth
Phone Number
Start_of_Journey_Date
Badges
Items
Item Name
Effect
Target
Pokemon Owned
Trainer_ID
Pokemon_Name
Level
Experience
Date Of Capture
Officials
Official_Name
Official_ID
Role
Location
Start_of_Service
Moves
Move_Name
Move_type
Description
Power
Accuracy
Power Points
Effect_on_opponent
Pokemon_Names
Gym
Gym_Name
Type
Location
Gym_Leader_ID
Pokemon
Name
Pokemon_Type
Height
Weight
Weakness
EvolvesTo
Tournament
Name
Start_date
Location
End_Date
No_of_participants
No_of_Badges_required
Event
Event_ID
Location
Description
Result
Time
Encounter
Event_ID
Trainer_ID
Pokemon_Captured_Name
Trainer_Items
PokemonUsed

1st Normal Form:

We need to separate out the multi-valued and composite attributes into different tables in order to meet the requirements of 1st Normal Form.

- We broke down the name and phone number of trainers into separate tables as they are composite values. Also, we created a separate table for badges owned since it was a multi valued attribute.
- Each gym now has its location in a separate table Gym_Location.
- Each official has their name and location in a separate table Official_Name and Official_Location.
- Each event and tournament also has its location in a separate table Event_Location and Tournament_Location.
- Each pokemon now has its weaknesses and types stored in a separate tables Pokemon_Weakness and Pokemon_Type, since each pokemon can be of several types and also be weak to several types.
- Each move now has its types in a different table Types_Of_Move, since a move can be of multiple types.
- We broke down the Pokemon Names attribute of Moves into a different table Pokemons_with_this_move.
- Event now has a separate Event_Location table.
- There is now a separate table Pokemon_Evolves_To detailing the next possible evolutions of any pokemon.
- Tournament now has a separate Tournament_Location table.
- New tables have been created for items and pokemon used in each encounter in the Encounter table, Encounter_Pokemons and Encounter_Items_Used.
- Similarly, new tables have been created for items and pokemons used by each trainer in the Encounter Table, Battle_Trainer1_Pokemon_Used, Battle_Trainer2_Pokemon_Used, Battle_Trainer1_Items and Battle_Trainer2_Items.

Text


Pokemon_Weakness
Name
Weakness
P.K
Items
Item Name
Effect
Target
P.K.
Pokemon Owned
Trainer_ID
Pokemon_Name
Level
Experience
Date Of Capture
P.K. F.K.
P.K. F.K.
Pokemon
Name
Height
Weight
P.K.
Pokemon_Type
Type
Name
P.K F.K.
Moves
Move_Name
Move_type
Description
Power
Accuracy
Power Points
Effect_on_opponent
P.K.
Trainer
Trainer_ID
Gender
Date_of_Birth
Start_of_Journey
P.K.
Types Of Move
Move_Type
Move_Name
P.K F.K.
Pokemons_with_this_move
Move_Name
Pokemon_Name
P.K. F.K. P.K. F.K.
Pokemon_Evolves_To
Pokemon_Name
Evolves_To
P.K. F.K. P.K. F.K.
Trainer_Name
First
Middle
Last
Trainer_ID
P.K. F.K.
Gym
Gym_Name
Type
Gym_Leader_ID
P.K.
F.K.
Officials
Official_ID
Role
Start_of_service
P.K
Trainer_Number
Trainer_Id
Country Code
Number
P.K. F.K.
Official_Name
Official_ID
First
Middle
Last
P.K. F.K.
Trainer_Badges
Trainer_ID
Badges
P.K. F.K. P.K. F.K.
Gym_Location
Gym_Name
Region
City
Street
P.K. F.K.
Official_Location
Official_ID
Region
City
P.K F.K.
Event
Event_ID
Description
Result
Time_of_occurance
P.K.
Event_Location
Region
Street
City
Event_ID
F.K.
Tournament_Location
Name
Region
City
Street
P.K. F.K.
Tournament
Name
Start_Date
End_Date
No_of_participants
No_of_Badges_required
P.K.
Battle
Event_ID
Trainer1_ID
Trainer2_ID
P.K. F.K.
F.K.
F.K.
Encounter
Event_ID
Trainer_ID
Pokemon_Captured_Name
P.K. F.K.
F.K.
F.K.
Battle_Trainer1_Pokemon_Used
Pokemon_Used
Event_ID
P.K. F.K. P.K. F.K.
Battle_Trainer1_Items
Items
Event_Id
P.K F.K P.K F.K
Battle_Trainer2_Items
Items
Event_ID
P.K F.K P.K F.K
Encounter_Pokemons
Pokemon
Event_Id
F.K.
P.K F.K.
Encounter_Items_Used
Items
Event_Id
F.K.
P.K F.K.
Battle_Trainer2_Pokemon_Used
Pokemon_Used
Event_ID
P.K. F.K. P.K. F.K.

2nd Normal Form:

The schema formulated is also in 2nd normal form since all non-prime attributes are fully functionally dependent on the primary keys in every relation.

3rd Normal Form:

The schema is also in 3rd normal form since no non-prime attributes are transitively dependent on the primary key in every relation.