# Pokémon API Project

# Our Team: The Committers

**Bianca**

**Naim**

**Amber**

Amber: “DON’T FORGET TO COMMIT!”

Proud Moment: 115 commits

# Agenda

# OUR APPROACH

# Project MVPs

## MVP 1

Create models, controllers, components & repositories based on class diagrams

## MVP 2

Implement GET, POST, DELETE Requests & Tests

## Extension 1

Additional Derived Queries

## Extension 2

Implement battle and Level-Increase functionality

# Project Timeline

**Day 1:**

- Brainstormed ideas before finalising theme

- Created MVPs and Extensions

**Day 2 (Paired Programming):**

- Created ERM and Class Diagrams

- Created the basic building blocks of our projects using POJO

- Added in basic RESTful API Queries using SQL (H2) & Spring Boot

**Day 3:**

- Started working on  Battle function

- Tested our queries using PostMan

- Added in Derived Queries

**Day 4:**

- Finalised Battle Functions

- Attempted implementing Extension

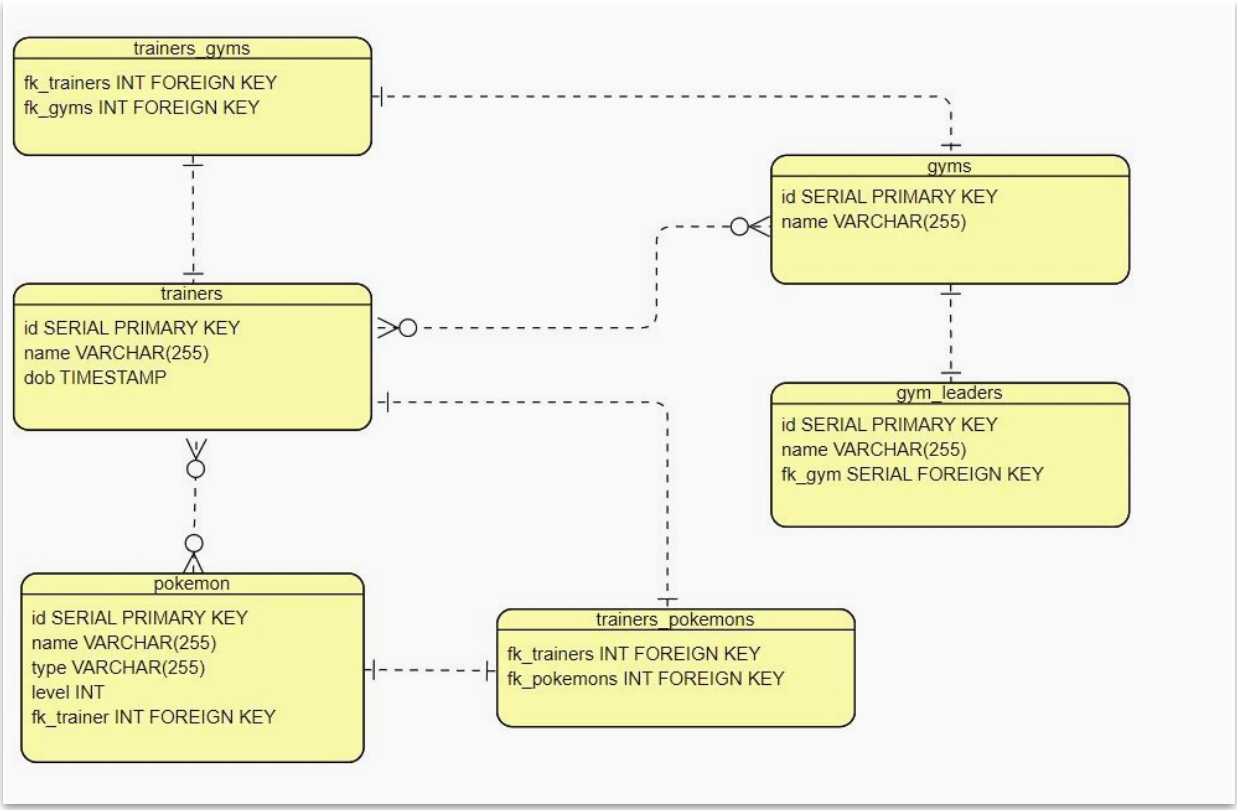
**Day 5:**

- Finalised our Project documentation in the ReadMe File

- Submitted our Pokemon Project

- Created Presentation Slides & Notes

# Entity-Relationship-Diagrams (ERD)



# H2 database tables and queries

Gym table:

| ID | NAME |
|---|---|
| 1 | Opelucid Gym |
| 2 | Driftveil Gym |
| 3 | Snowpoint Gym |
| 4 | Sunnyshore Gym |
| 5 | Mossdeep Gym |
| 6 | Striaton Gym |

Gym leader table:

| ID | NAME | GYM_ID |
|---|---|---|
| 1 | Iris | 1 |
| 2 | Clay | 2 |
| 3 | Bianca | 3 |
| 4 | Naim | 4 |
| 5 | Cece | 5 |
| 6 | Chris | 6 |

Trainer table:

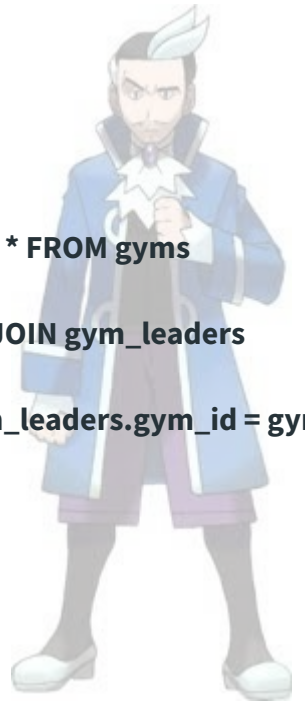
| ID | DOB | NAME |
|---|---|---|
| 1 | 2000-01-01 | Barry |
| 2 | 1998-05-06 | Lucas |
| 3 | 2010-02-03 | Chase |
| 4 | 1979-12-28 | Ash |
| 5 | 1992-10-10 | Brock |
| 6 | 2009-09-13 | Clemont |

Pokemon table:

| ID | LEVEL | NAME | TYPE |
|---|---|---|---|
| 1 | 50 | Machamp | 12 |
| 2 | 10 | Torkoal | 1 |
| 3 | 41 | Treeko | 2 |
| 4 | 25 | Krabby | 0 |
| 5 | 78 | Squirtle | 0 |
| 6 | 97 | Chimchar | 1 |
| 7 | 53 | Klinglang | 5 |
| 8 | 21 | Alakazam | 10 |
| 9 | 15 | Cinccino | 11 |
| 10 | 46 | Bulbasaur | 2 |

# One-to-one relationship between a gym and a gym leader

| ID | NAME | ID | NAME | GYM_ID |
|---|---|---|---|---|
| 1 | Opelucid Gym | 1 | Iris | 1 |
| 2 | Driftveil Gym | 2 | Clay | 2 |
| 3 | Snowpoint Gym | 3 | Bianca | 3 |
| 4 | Sunnyshore Gym | 4 | Naim | 4 |
| 5 | Mossdeep Gym | 5 | Cece | 5 |
| 6 | Striaton Gym | 6 | Chris | 6 |



```
SELECT * FROM gyms

INNER JOIN gym_leaders

ON gym_leaders.gym_id = gyms.id;
```

# Many-to-many relationship between trainers and pokemons

| POKEMON_ID | TRAINER_ID | ID | DOB | NAME | ID | LEVEL | NAME | TYPE |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2000-01-01 | Barry | 1 | 50 | Machamp | 12 |
| 1 | 2 | 2 | 1998-05-06 | Lucas | 1 | 50 | Machamp | 12 |
| 2 | 1 | 1 | 2000-01-01 | Barry | 2 | 10 | Torkoal | 1 |
| 3 | 1 | 1 | 2000-01-01 | Barry | 3 | 41 | Treeko | 2 |
| 3 | 2 | 2 | 1998-05-06 | Lucas | 3 | 41 | Treeko | 2 |
| 3 | 3 | 3 | 2010-02-03 | Chase | 3 | 41 | Treeko | 2 |
| 4 | 2 | 2 | 1998-05-06 | Lucas | 4 | 25 | Krabby | 0 |
| 4 | 4 | 4 | 1979-12-28 | Ash | 4 | 25 | Krabby | 0 |
| 5 | 5 | 5 | 1992-10-10 | Brock | 5 | 78 | Squirtle | 0 |
| 5 | 6 | 6 | 2009-09-13 | Clemont | 5 | 78 | Squirtle | 0 |
| 6 | 3 | 3 | 2010-02-03 | Chase | 6 | 97 | Chimchar | 1 |
| 6 | 4 | 4 | 1979-12-28 | Ash | 6 | 97 | Chimchar | 1 |
| 7 | 5 | 5 | 1992-10-10 | Brock | 7 | 53 | Klinglang | 5 |
| 8 | 6 | 6 | 2009-09-13 | Clemont | 8 | 21 | Alakazam | 10 |
| 9 | 1 | 1 | 2000-01-01 | Barry | 9 | 15 | Cinccino | 11 |
| 9 | 2 | 2 | 1998-05-06 | Lucas | 9 | 15 | Cinccino | 11 |
| 9 | 4 | 4 | 1979-12-28 | Ash | 9 | 15 | Cinccino | 11 |
| 10 | 3 | 3 | 2010-02-03 | Chase | 10 | 46 | Bulbasaur | 2 |

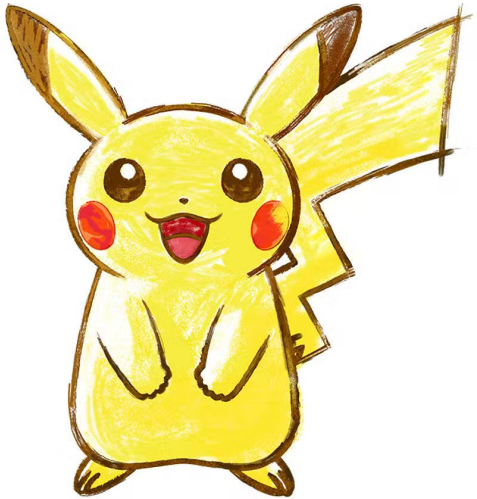
```
SELECT * FROM trainers_pokemons

INNER JOIN trainers

ON trainers.id = trainers_pokemons.trainer_id

INNER JOIN pokemons

ON pokemons.id = trainers_pokemons.pokemon_id;
```

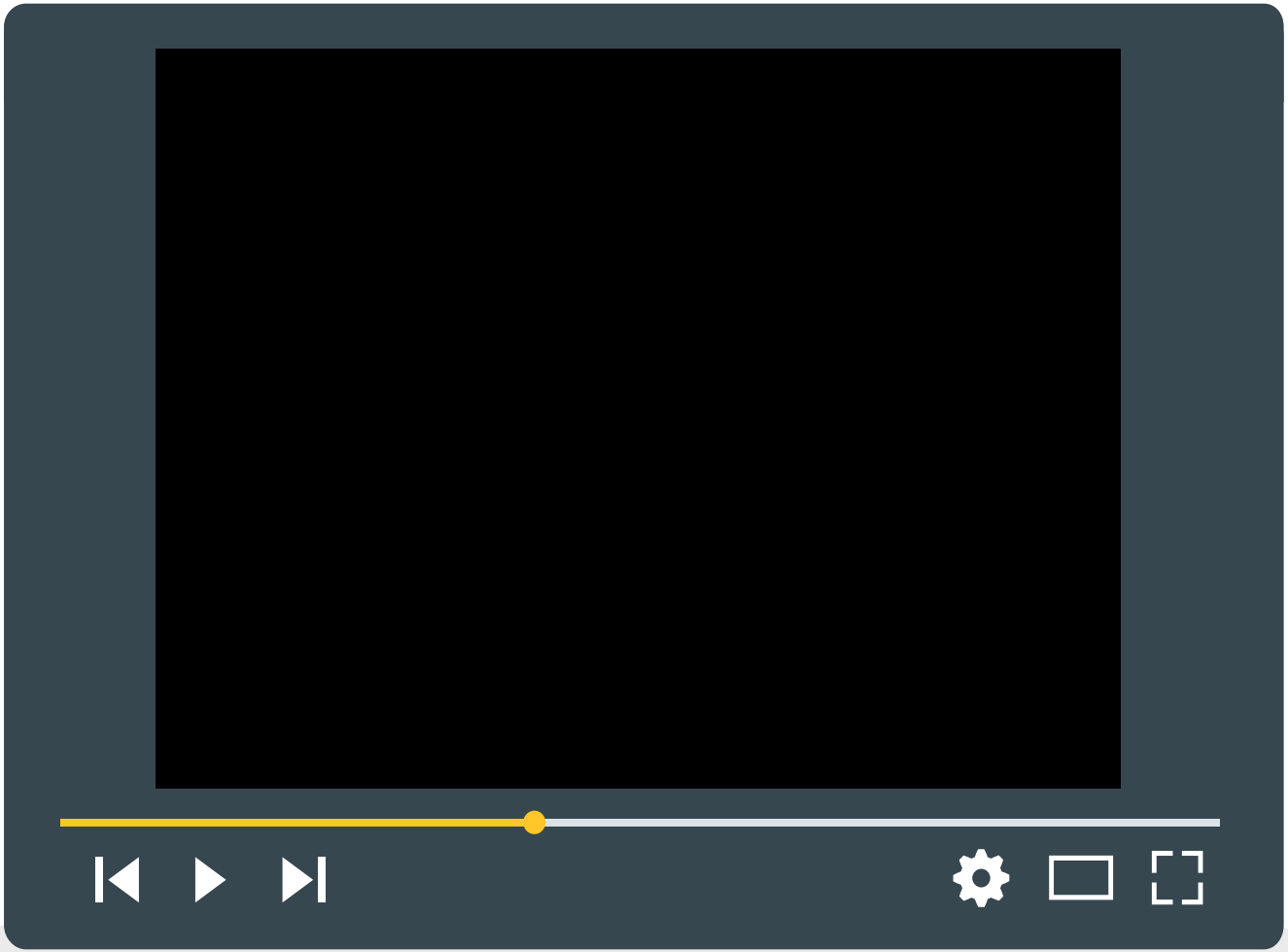
# Many-to-many relationship between gyms and trainers

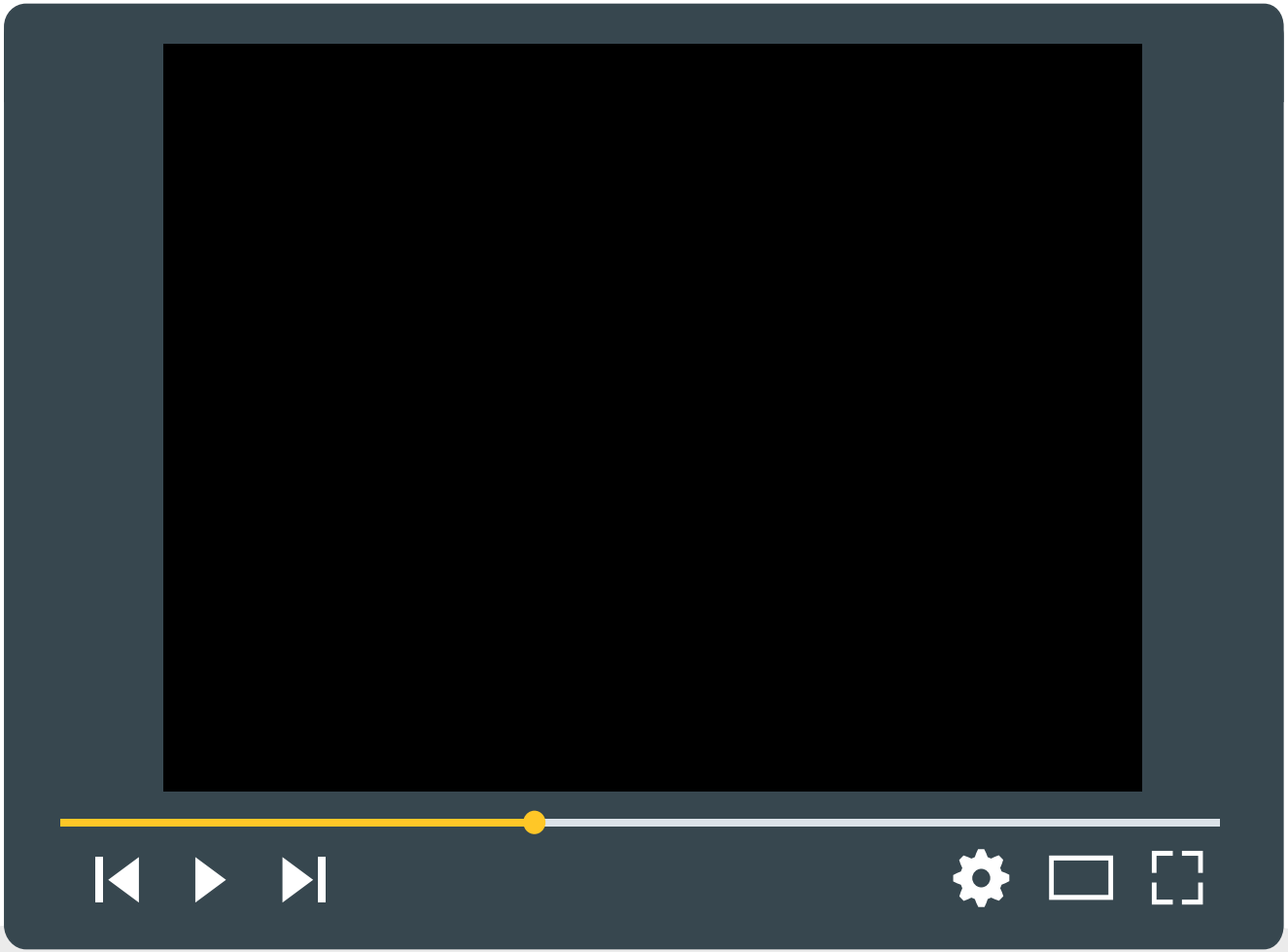
```
SELECT * FROM trainers_gyms

INNER JOIN trainers

ON trainers.id = trainers_gyms.trainer_id

INNER JOIN gyms

ON gyms.id = trainers_gyms.gym_id;
```

| GYM_ID | TRAINER_ID | ID | DOB | NAME | ID | NAME |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2000-01-01 | Barry | 1 | Opelucid Gym |
| 1 | 5 | 5 | 1992-10-10 | Brock | 1 | Opelucid Gym |
| 2 | 1 | 1 | 2000-01-01 | Barry | 2 | Driftveil Gym |
| 2 | 2 | 2 | 1998-05-06 | Lucas | 2 | Driftveil Gym |
| 2 | 3 | 3 | 2010-02-03 | Chase | 2 | Driftveil Gym |
| 3 | 1 | 1 | 2000-01-01 | Barry | 3 | Snowpoint Gym |
| 3 | 2 | 2 | 1998-05-06 | Lucas | 3 | Snowpoint Gym |
| 3 | 4 | 4 | 1979-12-28 | Ash | 3 | Snowpoint Gym |
| 4 | 1 | 1 | 2000-01-01 | Barry | 4 | Sunnyshore Gym |
| 4 | 2 | 2 | 1998-05-06 | Lucas | 4 | Sunnyshore Gym |
| 4 | 3 | 3 | 2010-02-03 | Chase | 4 | Sunnyshore Gym |
| 5 | 1 | 1 | 2000-01-01 | Barry | 5 | Mossdeep Gym |
| 5 | 4 | 4 | 1979-12-28 | Ash | 5 | Mossdeep Gym |
| 5 | 5 | 5 | 1992-10-10 | Brock | 5 | Mossdeep Gym |
| 5 | 6 | 6 | 2009-09-13 | Clemont | 5 | Mossdeep Gym |
| 6 | 1 | 1 | 2000-01-01 | Barry | 6 | Striaton Gym |
| 6 | 2 | 2 | 1998-05-06 | Lucas | 6 | Striaton Gym |
| 6 | 3 | 3 | 2010-02-03 | Chase | 6 | Striaton Gym |
| 6 | 4 | 4 | 1979-12-28 | Ash | 6 | Striaton Gym |
| 6 | 5 | 5 | 1992-10-10 | Brock | 6 | Striaton Gym |
| 6 | 6 | 6 | 2009-09-13 | Clemont | 6 | Striaton Gym |

# Demo

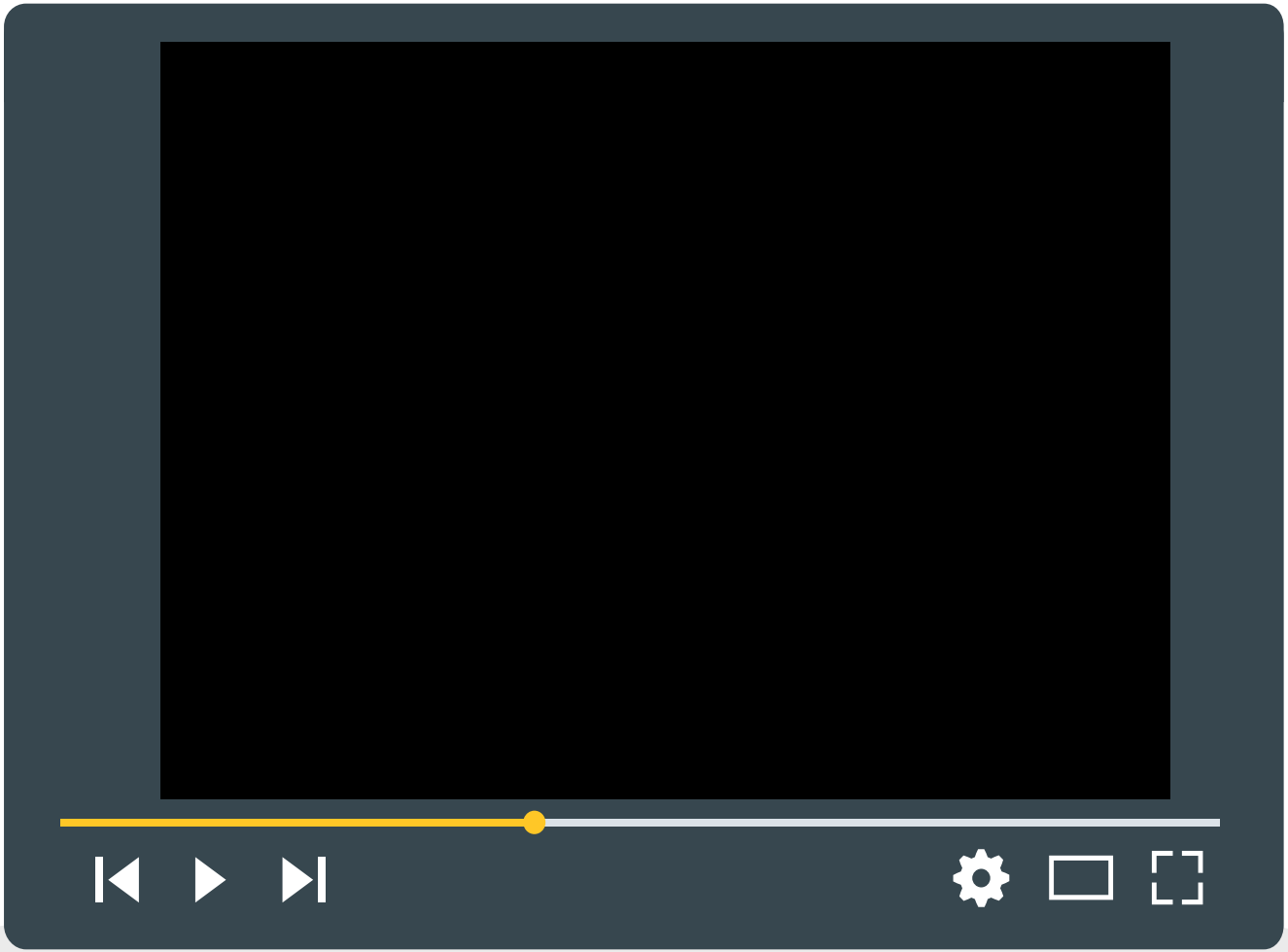






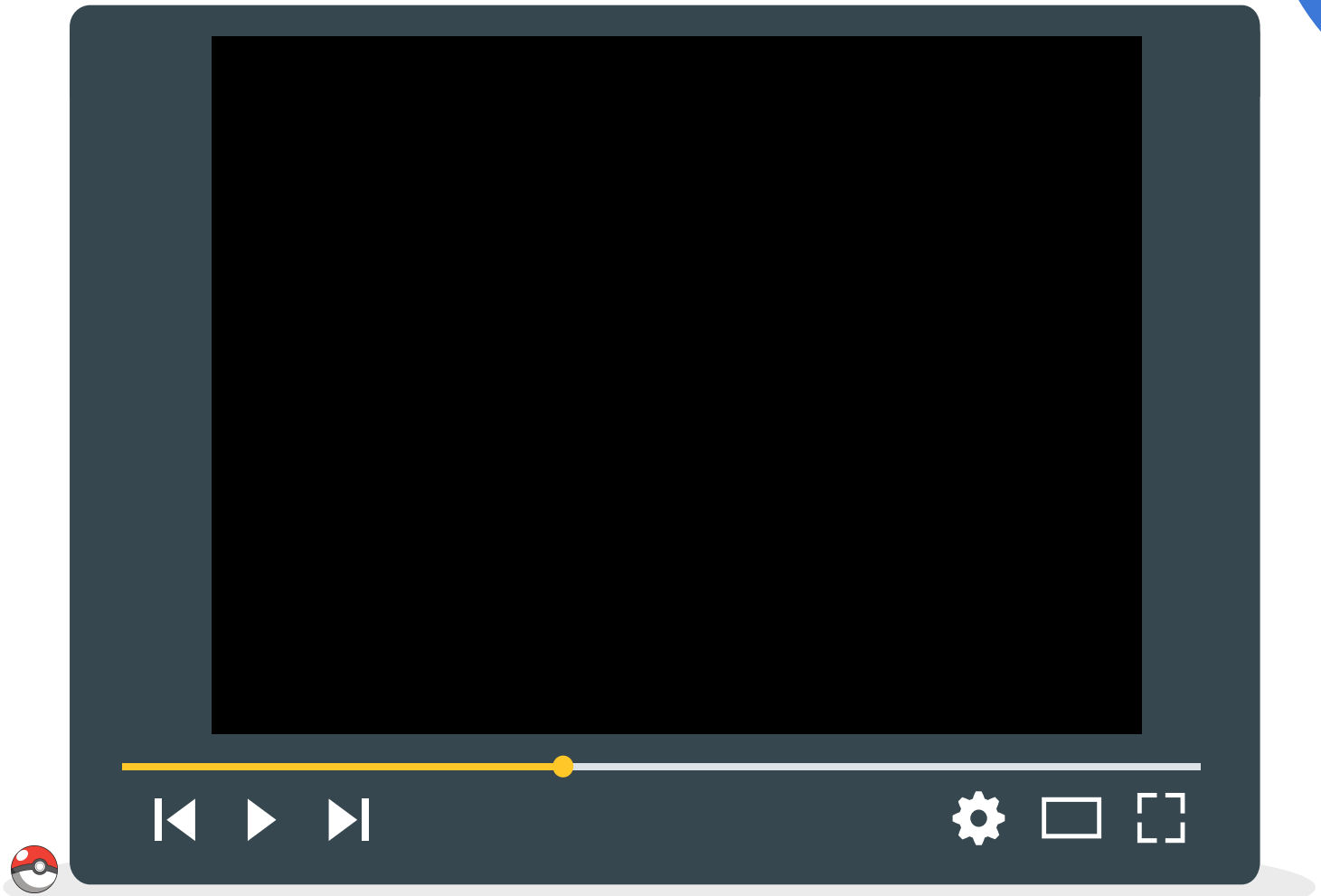
# PUT request

```java
// CHANGE: ADD POKEMON
@PutMapping("/addPokemon/{id_trainer}/{id_pokemon}")
public ResponseEntity<Trainer> addPokemonInTrainer(@PathVariable("id_trainer") Long id_trainer,
                                                   @PathVariable("id_pokemon") Long id_pokemon) {
    var found = trainerRepository.findById(id_trainer);
    Trainer trainerChange = found.get();
    if (trainerChange
            .getPokemons()
            .stream()
            .filter(pok -> pok.getId() == id_pokemon)
            .findAny()
            .isPresent()) {
       return new ResponseEntity(trainerRepository.findById(id_trainer).get(),
                                                   HttpStatus.ALREADY_REPORTED);
    }
    trainerChange.addPokemon(pokemonRepository
                    .findAll()
                    .stream()
                    .filter(pok -> pok.getId() == id_pokemon)
                   .findAny().get()
    );
    return new ResponseEntity(trainerChange, found.isEmpty() ? HttpStatus.NOT_FOUND : HttpStatus.OK);
}
```

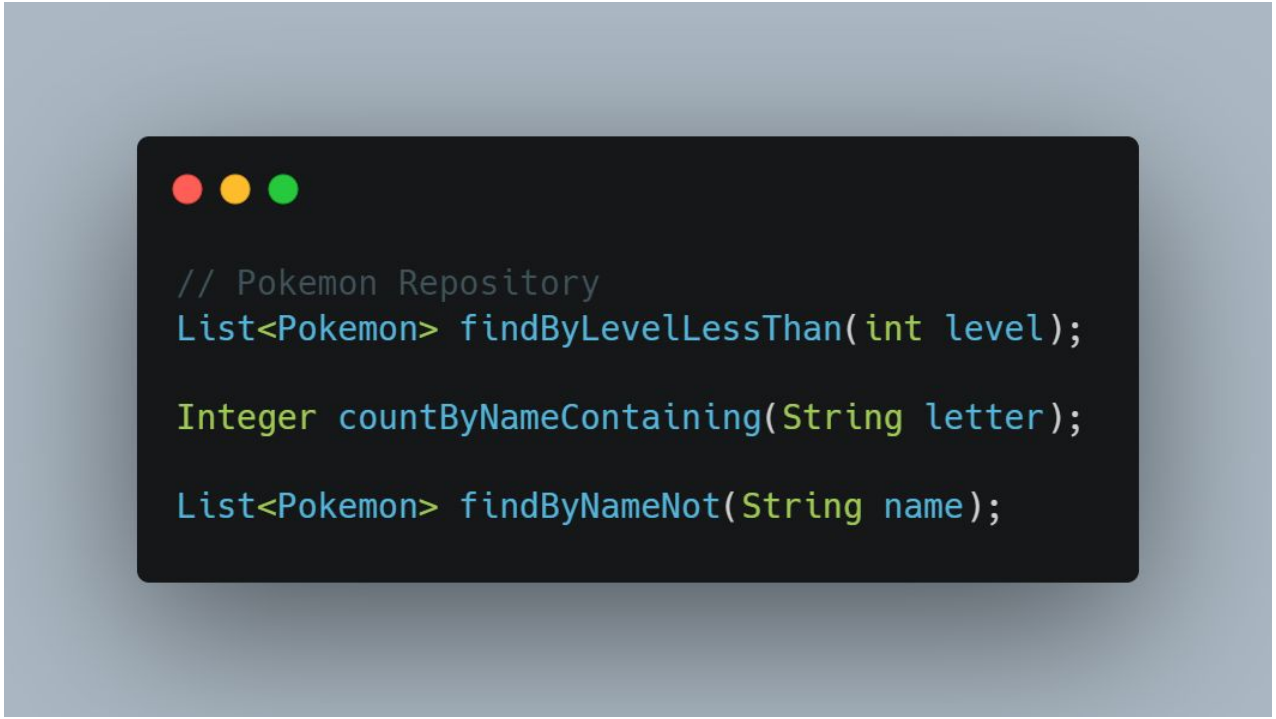


# Battle: calculate level increase method

```java
public int calculateLevelIncrease(Pokemon[] pokemons, boolean result) {
        int level1 = pokemons[0].getLevel();
        int level2 = pokemons[1].getLevel();
        int increase;
        // Trainer 1 is the winner
        if (result == true) {
            if (level1 >= level2) {
                increase = (level1 - level2) / 4;
            } else {
                increase = (level2 - level1) / 2;
            }
        }
        // Trainer 2 is the winner
        else {
            if (level1 >= level2) {
                increase = (level1 - level2) / 2;
            } else {
                increase = (level2 - level1) / 4;
            }
        }
        return increase;
    }
```



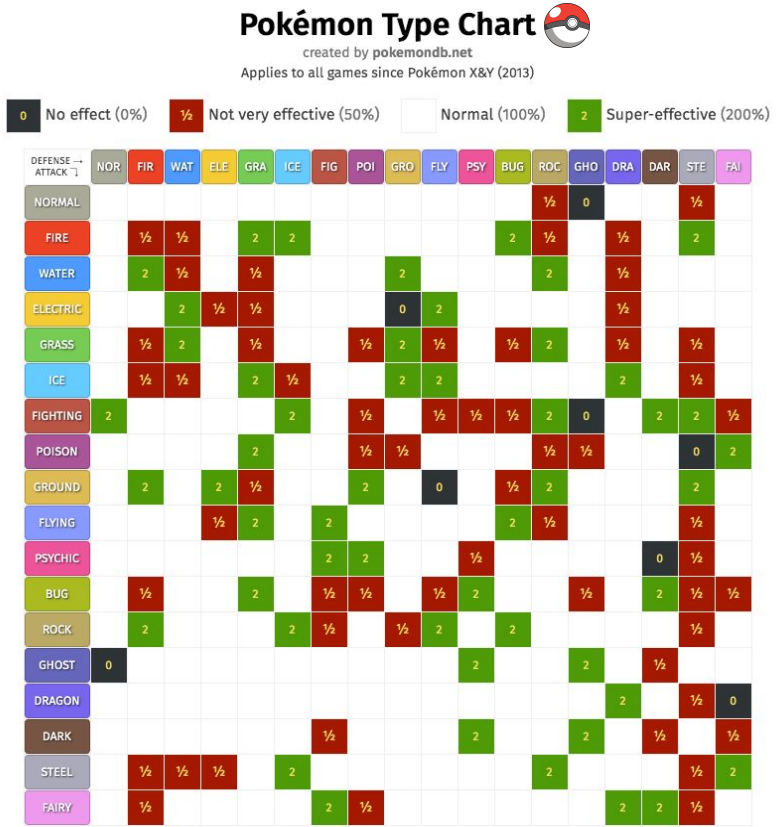
# Derived queries



```
// Pokemon Repository
List<Pokemon> findByLevelLessThan(int level);

Integer countByNameContaining(String letter);

List<Pokemon> findByNameNot(String name);
```

# Improvements & Issues

- Automate pokemon selection for battle
- Give pokemon abilities, multiple types
- Add method to determine winner and evolving at a certain level
- Add Exceptions



**Before we open to questions,**
**we have one for you!**

# The answer is 15!!!

*Did you notice how many Pokeballs we hid throughout our presentation?*

# THANK YOU

Any Questions?