University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___4____

Date: _____6th Aug 2025_____

Group Number: ____36____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Dikpaal Patel	37647864	dikpaal	dikpaalpatel123@gmail.com
Anthony Lu	35658681	alu34	anthonylu7678@gmail.com
Graydon Strachan	37275377	gstracha	glstrachan@outlook.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

FINAL PROJECT DESCRIPTION

1) A short description of the final project, and what it accomplished.

-> Our final project is a Pokemon Game Management Lookup System, built using JavaScript and OracleDB relational database. The system models key aspects of the Pokemon game universe, including Pokemon species, individual Pokemon, moves, abilities, trainers, items, and locations. It enables efficient exploration of in-game relationships such as move learnability, species, locations, and trainer-owned Pokemon through a web-based interface. This provided players with a strategic planning tool to optimize their experience. The project successfully demonstrates real-world application of relational database design, normalization, and query optimization, while integrating a clean user interface and backend logic.

2) A description of how your final schema differed from the schema you turned in. If the final schema differed, explain why. Note that turning in a final schema that's different from what you planned is fine, we just want to know what changed and why.

->

- In the 'Pokemon_3' table, the 'level' column was renamed to 'pokemon_level' and a 'UNIQUE' constraint was added. This change was made because 'level' is a reserved keyword.
- The data type of the 'difficulty' column in both 'Gym_Leader_1' and 'Gym_Leader_2' tables was changed from 'VARCHAR(10)' to 'INTEGER' because it is a more efficient way to represent difficulty.
- The 'ON DELETE CASCADE' constraint was added to the foreign key references in the 'Player', 'Gym_Leader_1', 'Pokemon_Has_Learned_Move', and 'Player_Owns_Badge' tables. This was in order to maintain relational integrity.
- In the 'Pokemon_1' table, 'DEFAULT 0' constraints were added to the 'attack_IV', 'defense_IV', 'speed_IV', and 'ability_id' columns. This change ensures that when a new Pokemon is created, these fields will have a default value if one is not explicitly provided.

3) A copy of the schema and screenshots that show what data is present in each relation after the SQL initialization script is run.

Schema:

Entities and One-to-Many-Relationships:

Pokemon_1(**pokedex**: INTEGER, pokemon_id: INTEGER, name: VARCHAR(20), **total_XP**: INTEGER, **nature**: VARCHAR(12), HP_IV: INTEGER, attack_IV: INTEGER, defense_IV: INTEGER, speed_IV: INTEGER, **ability_id**: INTEGER, **trainer_id**: INTEGER);

- trainer_id can be null
- FOREIGN KEY (pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (total_XP) REFERENCES Pokemon_3(total_XP)
- FOREIGN KEY (nature) REFERENCES Pokemon_2(nature)
- FOREIGN KEY (ability_id) REFERENCES Ability(ability_id)
- FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)

Pokemon_2(nature: VARCHAR(12), stat_increased: VARCHAR(12), stat_decreased: VARCHAR(12))

Pokemon_3(total_XP: INTEGER, pokemon_level: INTEGER)

Ability(ability_id: INTEGER, name: VARCHAR(255), description: VARCHAR(1000));

- name is a candidate key

Species(pokedex: INTEGER, name: VARCHAR(12), description:VARCHAR(1000));

- name is a candidate key

Egg_Group(egg_group_id: INTEGER, name: VARCHAR(30), description: VARCHAR(1000));

- name is a candidate key

Item_1(name: VARCHAR(255), description: VARCHAR(1000), **price**: INTEGER, **location_name**: VARCHAR(40));

- price and location_name can be null
- FOREIGN KEY (price) REFERENCES Item_2(price)
- FOREIGN KEY (location_name) REFERENCES Location(name)

Item_2(price: INTEGER, rarity: VARCHAR(10))

- rarity cannot be null here because price cannot be null

Type(name: VARCHAR(9), colour: VARCHAR(50));

- colour is a candidate key

Location(name: VARCHAR(40), climate: VARCHAR(255), terrain_type: VARCHAR(255))

Trainer(trainer_id: INTEGER, name: VARCHAR(255), **location_name**: VARCHAR(40));
- location_name can be null
- FOREIGN KEY (location_name) REFERENCES Location(name)

Gym_Leader_1(**trainer_id**: INTEGER, **difficulty**: VARCHAR(10), **specialty_type_name**: VARCHAR(9), **badge_index**: INTEGER)
- badge_index is unique
- badge_index is a candidate key
- FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)
- FOREIGN KEY (difficulty) REFERENCES Gym_Leader_2(difficulty)
- FOREIGN KEY (specialty_type_name) REFERENCES Type(name)
- FOREIGN KEY (badge_index) REFERENCES Badge(badge_index)

Gym_Leader_2(difficulty: VARCHAR(10), cash_reward: INTEGER)

Player(**trainer_id**: INTEGER, money: INTEGER)
- FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)

Move(move_id: INTEGER, name: VARCHAR(50), power: INTEGER, pp: INTEGER, accuracy: INTEGER, description: VARCHAR(1000), **type_name**: VARCHAR(9))
- name is a candidate key
- FOREIGN KEY (type_name) REFERENCES Type(name)

Badge(badge_index: INTEGER, name: VARCHAR(15))
- name is a candidate key

Many-to-Many Relationships:

Species_Evolves_Into(**old_pokedex**: INTEGER, **new_pokedex**: INTEGER, evolution_level: INTEGER);
- evolution_level can be null
- FOREIGN KEY (old_pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (new_pokedex) REFERENCES Species(pokedex)

Pokemon_Has_Learned_Move(**pokedex**: INTEGER, **pokemon_id**: INTEGER, **move_id**: INTEGER)

- FOREIGN KEY (pokedex, pokemon_id) REFERENCES Pokemon(pokedex, pokemon_id)
- FOREIGN KEY (move_id) REFERENCES Move(move_id)

Species_Has_Type(**pokedex**: INTEGER, **type_name**: VARCHAR(9));

- FOREIGN KEY (pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (type_name) REFERENCES Type(name)

Species_Can_Learn_Move(**pokedex**: INTEGER, **move_id**: INTEGER)

- FOREIGN KEY (pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (move_id) REFERENCES Move(move_id)

Species_Can_Have_Ability(**pokedex**: INTEGER, **ability_id**: INTEGER);

- FOREIGN KEY (pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (ability_id) REFERENCES Ability(ability_id)

Species_Located_In(**pokedex**: INTEGER, **location_name**: VARCHAR(40));

- FOREIGN KEY (pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (location_name) REFERENCES Location(name)

Species_Belongs_To_Egg_Group(**pokedex**: INTEGER, **egg_group_id**: INTEGER)

- FOREIGN KEY (pokedex) REFERENCES Species(pokedex)
- FOREIGN KEY (egg_group_id) REFERENCES Egg_Group(egg_group_id)

Player_Owns_Badge(**trainer_id**: INTEGER, **badge_index**: INTEGER)

- FOREIGN KEY (trainer_id) REFERENCES Player(trainer_id)
- FOREIGN KEY (badge_index) REFERENCES Badge(badge_index)

Higher quality versions of these images can be found in the repo

SPECIES LOCATED IN TABLE
(6 rows shown)

POKEDEX	LOCATION_NAME
1	Pallet Town
4	Pallet Town
7	Pallet Town
25	Vermilion City
25	Viridian City
26	Vermilion City

SPECIES TABLE
(11 rows shown)

POKEDEX	NAME	DESCRIPTION
1	Bulbasaur	It carries a seed on its back right from birth. As its body ...
2	Ivysaur	The bulb on its back grows as it absorbs nutrients. The bulb...
3	Venusaur	By spreading the broad petals of its flower and catching the...
4	Charmander	The flame on its tail shows the strength of its life-force. ...
5	Charmeleon	It is very hotheaded by nature, so it constantly seeks oppon...
6	Charizard	The flame inside its body burns hotter than 3,600 degrees Fa...
7	Squirtle	Its shell is soft immediately after it is born. In no time a...
8	Wartortle	It often hides in water to stalk unwary prey. While swimming...
9	Blastoise	It has jet nozzles on its shell. This impressive Pokémon use...
25	Pikachu	When several of these Pokémon gather, their electricity can ...
26	Raichu	Its tail discharges electricity into the ground, protecting ...

TRAINER TABLE
(13 rows shown)

TRAINER_ID	NAME	LOCATION_NAME
1	Ash	Pallet Town
2	Gary	NULL
3	Brock	Pewter City
4	Misty	Cerulean City
5	Lt. Surge	Vermilion City
6	Erika	Celadon City
7	Koga	Fuchsia City
8	Sabrina	Saffron City
9	Blaine	Route 1
10	Giovanni	NULL
11	John	Pallet Town
12	Joe	NULL
13	Amanda	NULL

TYPE TABLE
(18 rows shown)

NAME	COLOUR
Bug	Olive
Dark	Black
Dragon	Dark Blue
Electric	Yellow
Fairy	Light Pink
Fighting	Brown
Fire	Red
Flying	Sky Blue
Ghost	Indigo
Grass	Green
Ground	Earth
Ice	Light Blue
Normal	Tan
Poison	Purple
Psychic	Pink
Rock	Gray
Steel	Silver
Water	Blue

ABILITY TABLE
(8 rows shown)

ABILITY_ID	NAME	DESCRIPTION
2	Static	Contact with the Pokémon may cause paralysis.
4	Overgrow	Powers up Grass-type moves when the Pokémon's HP is low.
5	Blaze	Powers up Fire-type moves when the Pokémon's HP is low.
6	Torrent	Powers up Water-type moves when the Pokémon's HP is low.
8	Chlorophyll	Boosts the Pokémon's Speed stat in harsh sunlight.
9	Solar Power	Boosts the Sp. Atk stat in harsh sunlight, but HP decreases ...
10	Rain Dish	The Pokémon gradually regains HP in rain.
11	Lightning Rod	Draws in all Electric-type moves to boost its Sp. Atk stat.

BADGE TABLE
(6 rows shown)

BADGE_INDEX	NAME
1	Boulder Badge
2	Cascade Badge
3	Thunder Badge
4	Rainbow Badge
5	Soul Badge
6	Marsh Badge

POKEMON DATABASE - TABLE OVERVIEW

TABLE NAME	ROWS	COLUMNS
ABILITY	8	3
BADGE	6	2
EGG_GROUP	15	3
GYM_LEADER_1	6	4
GYM_LEADER_2	5	2
ITEM_1	6	4
ITEM_2	6	2
LOCATION	9	3
MOVE	10	7
PLAYER	5	2
PLAYER_OWNS_BADGE	6	2
POKEMON_1	12	11
POKEMON_2	16	3
POKEMON_3	45	2
POKEMON_HAS_LEARNED_MOVE	14	3
SPECIES	11	3
SPECIES_BELONGS_TO_EGG_GROUP	22	2
SPECIES_CAN_HAVE_ABILITY	22	2
SPECIES_CAN_LEARN_MOVE	26	2
SPECIES_EVOLVES_INTO	7	3
SPECIES_HAS_TYPE	15	2
SPECIES_LOCATED_IN	6	2
TRAINER	13	3
TYPE	18	2

EGG GROUP TABLE
(15 rows shown)

EGG_GROUP_ID	NAME	DESCRIPTION
1	Mineral	Pokemon in this group are inorganic in nature
2	Amorphous	Pokemon in this group are amorphous, having no definite form
3	Grass	Pokemon in this group are plantlike in appearance
4	Water 3	Pokemon in this group resemble aquatic invertebrates
5	Water 2	Pokemon in this group are piscine (fishlike) in appearance
6	Water 1	Pokemon in this group are amphibious in nature
7	Bug	Pokemon in this group are insectoid (bug-like) in appearance
8	Dragon	Pokemon in this group are reptilian or draconic in appearanc...
9	Flying	Pokemon in this group are avian (birdlike) in appearance
10	Field	The largest group, Pokemon here are terrestrial in nature
11	Human-Like	Pokemon in this group are fully bipedal humanoids
12	Fairy	Pokemon in this group are petite and considered very cute
13	Monster	Pokemon in this group are saurian/kaiju-like in appearance a...
14	Ditto	Ditto is the only Pokemon in this group, capable of breeding...
15	No Eggs Discovered	Pokemon in this group are unable to breed

GYM LEADER 1 TABLE
(6 rows shown)

TRAINER_ID	DIFFICULTY	SPECIALTY_TYPE_NAME	BADGE_INDEX
3	1	Rock	1
4	2	Water	2
5	3	Electric	3
6	4	Grass	4
7	5	Poison	5
8	5	Psychic	6

GYM LEADER 2 TABLE
(5 rows shown)

DIFFICULTY	CASH_REWARD
1	1000
2	2000
3	3000
4	5000
5	10000

ITEM 1 TABLE
(6 rows shown)

NAME	DESCRIPTION	PRICE	LOCATION_NAME
Bicycle	A folding Bicycle that enables a rider to get aro	0	Cerulean City
Poké Ball	A ball thrown to catch a wild Pokémon. It has a	100	Viridian City
Potion	A spray-type medicine for treating wounds. It re	200	Viridian City
Rare Candy	A candy that is packed with energy. When cons	2500	NULL
Super Potion	A spray-type medicine for treating wounds. It re	600	Cerulean City
TM28 - Dig	A TM that teaches the move Dig to a compatibl	1000	Vermilion City

ITEM 2 TABLE
(6 rows shown)

PRICE	RARITY
0	Quest
100	Common
200	Common
600	Uncommon
1000	Rare
2500	Very Rare

LOCATION TABLE
(9 rows shown)

NAME	CLIMATE	TERRAIN_TYPE
Celadon City	Temperate	Urban
Cerulean City	Coastal	Beach
Fuchsia City	Tropical	Swamp
Pallet Town	Temperate	Grassland
Pewter City	Mountainous	Rocky
Route 1	Temperate	Grassland
Saffron City	Temperate	Urban
Vermilion City	Coastal	Port
Viridian City	Temperate	Urban

MOVE TABLE
(10 rows shown)

MOVE_ID	NAME	POWER	PP	ACCURACY	DESCRIPTION	TYPE_NAME
1	Tackle	40	35	100	A physical attack in which the user char	Normal
2	Growl	0	40	100	The user growls in an endearing way, m	Normal
3	Vine Whip	45	25	100	The target is struck with slender, whip-l	Grass
4	Ember	40	25	100	The target is attacked with small flames	Fire
5	Flamethrower	90	15	100	The target is scorched with an intense b	Fire
6	Water Gun	40	25	100	The target is blasted with a forceful jet	Water
7	Hydro Pump	110	5	80	The target is blasted by a huge volume	Water
8	Thunder Shock	40	30	100	A jolt of electricity is hurled at the targe	Electric
9	Quick Attack	40	30	100	The user lunges at the target at a speed	Normal
10	Double Team	0	15	0	By moving rapidly, the user makes illus	Normal

PLAYER OWNS BADGE TABLE
(6 rows shown)

TRAINER_ID	BADGE_INDEX
1	1
1	2
1	3
1	4
2	1
2	2

PLAYER TABLE
(5 rows shown)

TRAINER_ID	MONEY
1	5000
2	3000
11	5000
12	3000
13	5000

POKEMON 1 TABLE
(12 rows shown, 10 of 11 columns shown)

POKEDEX	POKEMON_ID	NAME	TOTAL_XP	NATURE	HP_IV	ATTACK_IV	DEFENSE_IV	SPEED_IV	ABILITY_ID
1	1	Bulbasaur	1000	Modest	25	15	20	18	4
2	1	Ivysaur	3000	Calm	28	18	25	20	4
3	1	Venusaur	6000	Bold	31	20	31	22	4
4	1	Charmander	1000	Jolly	22	25	18	28	5
5	1	Charmeleon	3000	Adamant	28	30	20	25	5
6	1	Charizard	6000	Adamant	31	31	22	28	5
7	1	Squirtle	1000	Bold	24	18	28	20	6
8	1	Wartortle	3000	Relaxed	28	20	30	18	6
9	1	Blastoise	6000	Modest	31	15	31	25	6
25	1	Pikachu	3000	Timid	20	18	15	31	2
25	2	Pikachu	1000	Relaxed	1	31	9	6	11
26	1	Raichu	4000	Jolly	25	25	20	31	2

POKEMON 2 TABLE
(16 rows shown)

NATURE	STAT_INCREASED	STAT_DECREASED
Adamant	Attack	Sp. Atk
Bold	Defense	Attack
Brave	Attack	Speed
Calm	Sp. Def	Attack
Careful	Sp. Def	Sp. Atk
Hasty	Speed	Defense
Impish	Defense	Sp. Atk
Jolly	Speed	Sp. Atk
Lonely	Attack	Defense
Mild	Sp. Atk	Defense
Modest	Sp. Atk	Attack
Naive	Speed	Sp. Def
Quiet	Sp. Atk	Speed
Rash	Sp. Atk	Sp. Def
Relaxed	Defense	Speed
Timid	Speed	Attack

POKEMON 3 TABLE
(20 of 45 rows shown)

TOTAL_XP	POKEMON_LEVEL
0	1
250	2
500	3
750	4
1000	5
1400	6
1800	7
2200	8
2600	9
3000	10
3600	11
4200	12
4800	13
5400	14
6000	15
6800	16
7600	17
8400	18
9200	19
10000	20

POKEMON HAS LEARNED MOVE TABLE
(14 rows shown)

POKEDEX	POKEMON_ID	MOVE_ID
1	1	1
1	1	2
1	1	3
4	1	1
4	1	2
4	1	4
7	1	1
7	1	2
7	1	6
25	1	1
25	1	8
25	2	9
26	1	8
26	1	9

SPECIES BELONGS TO EGG GROUP TABLE
(20 of 22 rows shown)

POKEDEX	EGG_GROUP_ID
1	3
1	13
2	3
2	13
3	3
3	13
4	8
4	13
5	8
5	13
6	8
6	13
7	6
7	13
8	6
8	13
9	6
9	13
25	10
25	12

SPECIES CAN HAVE ABILITY TABLE
(20 of 22 rows shown)

POKEDEX	ABILITY_ID
1	4
1	8
2	4
2	8
3	4
3	8
4	5
4	9
5	5
5	9
6	5
6	9
7	6
7	10
8	6
8	10
9	6
9	10
25	2
25	11

SPECIES CAN LEARN MOVE TABLE
(20 of 26 rows shown)

POKEDEX	MOVE_ID
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3
4	1
4	4
5	1
5	5
6	5
6	9
7	1
7	6
8	1
8	6
9	1

SPECIES EVOLVES INTO TABLE
(7 rows shown)

OLD_POKEDEX	NEW_POKEDEX	EVOLUTION_LEVEL
1	2	16
2	3	32
4	5	16
5	6	36
7	8	16
8	9	36
25	26	0

SPECIES HAS TYPE TABLE
(15 rows shown)

POKEDEX	TYPE_NAME
1	Grass
1	Poison
2	Grass
2	Poison
3	Grass
3	Poison
4	Fire
5	Fire
6	Fire
6	Flying
7	Water
8	Water
9	Water
25	Electric
26	Electric

4) A list of all SQL queries

setup_all.sql:

appService.js:

- 327:

```
SELECT COUNT(*) FROM Trainer
```

- 334:

```
SELECT COUNT(*) FROM Pokemon_1
```

- 370:

```
SELECT * FROM Trainer ORDER BY trainer_id
```

- 375:

```
SELECT p.trainer_id, t.name, p.money
FROM Player p, Trainer t
WHERE p.trainer_id = t.trainer_id
ORDER BY p.trainer_id
```

- 384:

```
SELECT p1.pokedex, p1.pokemon_id, p1.name, p3.pokemon_level, p1.nature,
p1.HP_IV, p1.attack_IV, p1.defense_IV, p1.speed_IV, p1.ability_id, p1.trainer_id
FROM Pokemon_1 p1, Pokemon_3 p3
WHERE p1.total_XP = p3.total_XP
ORDER BY p1.pokedex, p1.pokemon_id
```

- 394:

```
SELECT lm.pokedex, lm.pokemon_id, lm.move_id, p.name as pokemon_name,
m.name as move_name
FROM Pokemon_Has_Learned_Move lm, Pokemon_1 p, Move m
WHERE lm.pokedex = p.pokedex AND lm.pokemon_id = p.pokemon_id AND
lm.move_id = m.move_id
ORDER BY lm.pokedex, lm.pokemon_id, lm.move_id
```

- 402:

```
SELECT * FROM Species ORDER BY pokedex
```

- 406:

```
'SELECT move_id, name, type_name, power, pp, accuracy FROM Move ORDER BY
move_id'
```

- 410:

```
'SELECT * FROM Ability ORDER BY ability_id'
```

- 414:

```
'SELECT * FROM Pokemon_2 ORDER BY nature'
```

- 426:

```
INSERT INTO Trainer (trainer_id, name, location_name) VALUES (:trainerId, :name,
:locationName)
```

- 444:

```
INSERT INTO Player (trainer_id, money) VALUES (:trainerId, :money)
```

- 471:

```
INSERT INTO Pokemon_1
```

(pokedex, pokemon_id, name, total_XP, nature, HP_IV, attack_IV, defense_IV, speed_IV, ability_id, trainer_id)
VALUES
(:pokedex, :pokemon_id, :name, :total_XP, :nature, :HP_IV, :attack_IV, :defense_IV, :speed_IV, :ability_id, :trainer_id)

- 493:

```
INSERT INTO Pokemon_Has_Learned_Move (pokedex, pokemon_id, move_id)
VALUES (:pokedex, :pokemon_id, :move_id)
```

- 511:

```
DELETE FROM Trainer WHERE trainer_id = :trainerId
```

- 529:

```
DELETE FROM Pokemon_1 WHERE pokedex = :pokedex AND pokemon_id = :pokemon_id
```

- 548:

```
DELETE FROM Pokemon_Has_Learned_Move WHERE pokedex = :pokedex AND pokemon_id = :pokemon_id AND move_id = :move_id
```

- 592:

```
SELECT 1 FROM Ability WHERE ability_id = :abilityId
```

- 652:

```
SELECT 1 FROM Pokemon_Has_Learned_Move WHERE pokedex = :pokedex AND pokemon_id = :pokemonId AND move_id = :moveId
```

- 664:

```
SELECT 1 FROM Move WHERE move_id = :moveId
```

- 676:

```
SELECT 1 FROM Species_Can_Learn_Move WHERE pokedex = :pokedex AND move_id = :moveId
```

- 688:

```
SELECT * FROM Pokemon_Has_Learned_Move WHERE pokedex = :pokedex AND pokemon_id = :pokemonId AND move_id = :moveId
```

- 700:

```
UPDATE Pokemon_Has_Learned_Move
 SET move_id = :newMoveId
 WHERE pokedex = :pokedex AND pokemon_id = :pokemonId AND move_id = :oldMoveId
```

- 722:

```
SELECT 1 FROM Trainer WHERE trainer_id = :id
```

- 735:

```
SELECT 1 FROM Pokemon_1 WHERE pokedex = :pokedex AND pokemon_id = :pokemonId
```

- 789:

UPDATE ${validTable} SET ${setClauses.join(', ')} WHERE ${whereClause}

- 806-833:

```
SELECT p1.pokedex, p1.pokemon_id, p1.name, p3.pokemon_level, p1.nature,
                p1.HP_IV, p1.attack_IV, p1.defense_IV, p1.speed_IV, p1.ability_id,
p1.trainer_id
         FROM Pokemon_1 p1, Pokemon_3 p3
         WHERE p1.total_XP = p3.total_XP
```

- 847:

```
SELECT table_name FROM user_tables ORDER BY table_name
```

- 872:

```
SELECT column_name FROM user_tab_columns WHERE table_name = :tableName
ORDER BY column_name
```

- 893:

```
SELECT ${validAttributes.join(', ')} FROM ${validTableName}
```

- 906-919:

```
SELECT s.pokedex, s.name as species_name, s.description, l.name as location_name,
l.climate, l.terrain_type
         FROM Species s, Species_Located_In sli, Location l
         WHERE s.pokedex = sli.pokedex AND sli.location_name = l.name
```

- 931:

```
SELECT DISTINCT name FROM Location ORDER BY name
```

- 943-955:

```
SELECT t.trainer_id, t.name as trainer_name, p.name as pokemon_name,
p.defense_IV
         FROM Pokemon_1 p, Trainer t
         WHERE p.trainer_id = t.trainer_id
```

- 968:

```
SELECT t.trainer_id, t.name as trainer_name, COUNT(p.pokemon_id) as
pokemon_count,
                AVG(p.defense_IV) as avg_defense_iv, MAX(p.defense_IV) as
max_defense_iv
         FROM Trainer t
         LEFT JOIN Pokemon_1 p ON t.trainer_id = p.trainer_id
         GROUP BY t.trainer_id, t.name
         ORDER BY pokemon_count DESC, t.trainer_id
```

- 990:

```
SELECT t.trainer_id, t.name as trainer_name, MAX(p.total_XP) as highest_xp,
COUNT(p.pokemon_id) as pokemon_count
         FROM Trainer t, Pokemon_1 p
         WHERE t.trainer_id = p.trainer_id
```

```
        GROUP BY t.trainer_id, t.name
        HAVING COUNT(p.pokemon_id) >= :minPokemonCount
        ORDER BY highest_xp DESC
    - 1009-1029:
SELECT t.trainer_id, t.name as trainer_name,
COUNT(p.pokemon_id) as pokemon_count,
AVG(p.total_XP) as avg_xp,
MAX(p.total_XP) as max_xp,
MIN(p.total_XP) as min_xp
FROM Trainer t, Pokemon_1 p
WHERE t.trainer_id = p.trainer_id
    - 1043:
SELECT t.trainer_id, t.name as trainer_name,
COUNT(p.pokemon_id) as pokemon_count,
AVG(p.total_XP) as avg_xp
FROM Trainer t, Pokemon_1 p
WHERE t.trainer_id = p.trainer_id
GROUP BY t.trainer_id, t.name
HAVING AVG(p.total_XP) > (
  SELECT AVG(total_XP)
  FROM Pokemon_1
)
ORDER BY avg_xp DESC
    - 1067:
        SELECT trainer_stats.trainer_id,
            trainer_stats.trainer_name,
            trainer_stats.pokemon_count,
            trainer_stats.avg_trainer_xp,
            global_stats.overall_avg_xp
        FROM (
            SELECT t.trainer_id, t.name as trainer_name,
                COUNT(p.pokemon_id) as pokemon_count,
                AVG(p.total_XP) as avg_trainer_xp
            FROM Trainer t, Pokemon_1 p
            WHERE t.trainer_id = p.trainer_id
            GROUP BY t.trainer_id, t.name
        ) trainer_stats, (
            SELECT AVG(total_XP) as overall_avg_xp
            FROM Pokemon_1
        ) global_stats
```

```
        WHERE trainer_stats.avg_trainer_xp > global_stats.overall_avg_xp
        ORDER BY trainer_stats.avg_trainer_xp DESC
```

- 1102:

```
SELECT s.pokedex, s.name as species_name,
LISTAGG(sht.type_name, ', ') WITHIN GROUP (ORDER BY sht.type_name) as types
FROM Species s, Species_Has_Type sht
WHERE s.pokedex = sht.pokedex
GROUP BY s.pokedex, s.name
ORDER BY s.pokedex
```

- 1130:

```
        SELECT s.pokedex, s.name as species_name,
            LISTAGG(sht.type_name, ', ') WITHIN GROUP (ORDER BY
sht.type_name) as types
        FROM Species s, Species_Has_Type sht
        WHERE s.pokedex = sht.pokedex
        AND s.pokedex IN (
            SELECT sht2.pokedex
            FROM Species_Has_Type sht2
            WHERE sht2.type_name IN (${typeConditions})
            GROUP BY sht2.pokedex
            HAVING COUNT(DISTINCT sht2.type_name) = :typeCount
        )
        GROUP BY s.pokedex, s.name
        ORDER BY s.pokedex
```

- 1154:

```
SELECT name FROM Type ORDER BY name
```

- 1163-1178:

```
SELECT s.pokedex, s.name as species_name,
        COUNT(sht.type_name) as type_count,
        LISTAGG(sht.type_name, ', ') WITHIN GROUP (ORDER BY sht.type_name) as
        types
FROM Species s, Species_Has_Type sht
WHERE s.pokedex = sht.pokedex
GROUP BY s.pokedex, s.name
```