1. Project Discription

Our Pokemon Data Management System is designed to help trainers optimize their gameplay by allowing them to store, delete, and retrieve specific Pokemon data. In addition to basic data management, the system offers insights that support strategic decision-making. Users can view average attack values grouped by type name, identify type names with an average defense greater than 10, and find trainers who own at least one Pokémon from every category. Through these features, our system helps trainers make more informed choices and improve their overall game performance.

2. Schema Differences and Explanation

We changed the schema name from **Type** to **PokemonType** to improve readability.

3. SQL Query List and Locations in Code

<table>
<tr><th>Functionality</th><th>SQL Query Summary</th><th>Line(s)</th></tr>
<tr><td><b>Queries1: INSERT</b></td><td>Insert into PokemonTrains with FK check on TypeName
<pre>INSERT INTO PokemonTrains (PokemonID,
PokemonName, TypeName, PokemonGender, Ability,
TrainerID)
           VALUES (:id, :name, :type, :gender,
:ability, :trainer)</pre></td><td>appService.js
line: 174
link</td></tr>
<tr><td><b>Query2: Update</b></td><td>Updates any combination of non-PK fields: PokemonName, TypeName, PokemonGender, Ability, TrainerID.
<pre>UPDATE PokemonTrains
                     SET ${update.join(', ')}
                     WHERE PokemonID = :id</pre></td><td>appService.js,
line: 265
link</td></tr>
<tr><td><b>Query3: Delete</b></td><td>Delete a tuple from PokemonTrains by PokemonID
<pre>DELETE FROM PokemonTrains WHERE PokemonID = :id</pre></td><td>appService.js,
line: 287
link</td></tr>
<tr><td><b>Query4: Selection</b></td><td>Allowed to search for tuples using any number of AND/OR clauses and combinations of attributes in PokemonTrains</td><td>appService.js,
line: 310
link</td></tr>
</table>

	`SELECT ${selectClause} FROM PokemonTrains p JOIN PokemonType t ON p.TypeName = t.TypeName WHERE ${whereClause}`;	
Query5: Projection	Project any number of attributes from the PokemonTrains table `SELECT ${attributes} FROM PokemonTrains p JOIN PokemonType t ON p.TypeName = t.TypeName`)	appService.js, line: 331 link

Query6: Join	Join Trainer and PokemonTrains to find all Pokemon trained by a specific trainer
	SELECT t.TrainerID, t.TrainerName, p.PokemonID, p.PokemonName
	FROM Trainer t
	JOIN PokemonTrains p ON t.TrainerID = p.TrainerID
	WHERE t.TrainerID=:trainerID
		appService.js, line: 345
		link

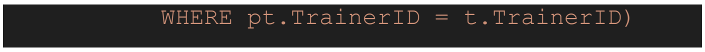
Query7: Aggregation With GROUP BY	Get average attack value grouped by TypeName
	SELECT p.TypeName, AVG(s.Attack) AS AvgAttack
	FROM PokemonTrains p
	JOIN Shows sh ON p.PokemonID = sh.PokemonID
	JOIN Stats s ON sh.StatsID = s.StatsID
	GROUP BY p.TypeName
		appService.js, line 365
		link

Query8: Aggregation With HAVING	Get TypeNames with average defense > 10
	SELECT p.TypeName, AVG(s.Defense) AS AvgDefense
	FROM PokemonTrains p
	JOIN Shows sh ON p.PokemonID = sh.PokemonID
		appService.js, line 383
		link

	JOIN Stats s ON sh.StatsID = s.StatsID GROUP BY p.TypeName HAVING AVG(s.Defense) > 10 FROM PokemonTrains p JOIN Shows sh ON p.PokemonID = sh.PokemonID JOIN Stats s ON sh.StatsID = s.StatsID GROUP BY p.TypeName HAVING AVG(s.Defense) > 10	

Query9: Nested Aggregation	Get trainer names whose Pokemon have above-average total stats
	SELECT t.TrainerName, AVG(s.HP + s.Attack + s.Defense + s.SpecialAttack + s.SpecialDefense + s.Speed) AS AvgTotalStats
	FROM Trainer t
	JOIN PokemonTrains pt ON t.TrainerID = pt.TrainerID
	JOIN Shows sh ON pt.PokemonID = sh.PokemonID
	JOIN Stats s ON sh.StatsID = s.StatsID
	GROUP BY t.TrainerName
	HAVING AVG(s.HP + s.Attack + s.Defense + s.SpecialAttack + s.SpecialDefense + s.Speed) > (
	SELECT AVG(HP + Attack + Defense + SpecialAttack + SpecialDefense + Speed)
	FROM Stats)
		appService.js, line: 402 link

Query10:Division	Get trainers who own at least one Pokemon from every category
	SELECT t.TrainerName
	FROM Trainer t
	WHERE NOT EXISTS (
	SELECT c.CategoryName FROM Category c
	MINUS
	SELECT b.CategoryName
	FROM BelongsTo b
	JOIN PokemonTrains pt ON b.PokemonID = pt.PokemonID
		appService.js, line: 424 link

	 `WHERE pt.TrainerID = t.TrainerID)`	

4. Acknowledgment

We used Grammarly to improve sentence fluency and correct syntax mistakes.

We based the initial structure of our project on the CPSC304 provided sample project.

5. Accidentally commit to using another account

Zhaowei Cheng: commit under zcheng32@student.ubc.ca

840df0c49ff22039a5ce424f7f1d076868583dd9
e1ef76ca44515c06e76cbb5f82eb9bbd91a8aca9
9ac42b9ce121e75c0d8ccbf0a5590b1c919678f7
f87acc2f7aaafed1d9ac7bdfe17c0f355ff79f34

Ruiyang Zhang: commit under rzhan101@student.ubc.ca

33cedfd6e6fc823f97e75c16b5437c9a301ad3ab
ddf75e66064fbfcbaa21aa5bddac46d43393c1c5
a79a389cfb0016051ef1de314623ff2ec9e7acf9
fa4e2db56bc14e086e18462936ff8d2e5272bb66
8f4d133115443121cc4eb59ae686c4beffdfd83f
5aac7cc8bbaf032e1a383ea0acedbdc8bfae783c
a8dc1ed90ea9a45f54e781e65676321657ae6377