FINA6204 CRN 19106 Data Analytics Assignment 3

Group Number: 9

Introduction:

This project focuses on implementing a relational database model to support analytics on a fictional game environment inspired by Pokémon. The goal was to design, populate, and query a structured dataset using MySQL and MySQL Workbench to extract meaningful insights about player behavior, product performance, and game mechanics.

Background:

In the context of modern data-driven gaming platforms, understanding user behavior and product interactions is essential for improving player engagement and balancing in-game elements. This project simulates a real-world gaming data environment by modeling various entities such as players, battles, sessions, and in-game purchases. Leveraging structured query language (SQL), we built a relational schema that allowed for scalable data analysis and business intelligence reporting.

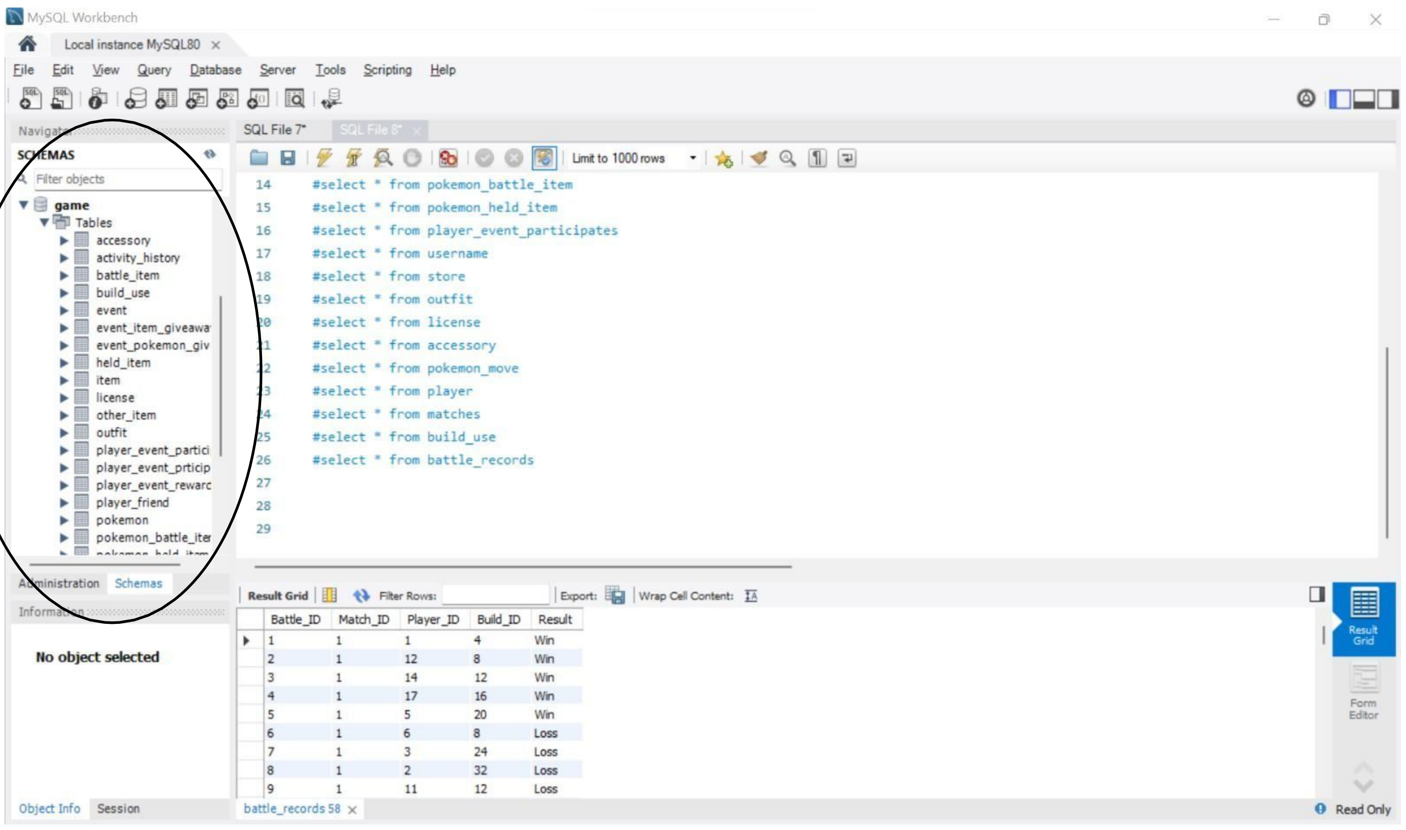
Objectives:

- To design and implement a normalized relational database schema for game-related datasets.
- To populate the schema with representative sample data across multiple entities.
- To write and execute analytical SQL queries that address specific business use cases such as:
    - Evaluating average player session duration and customer demographics
    - Identifying high-performing in-game builds and top-selling Pokémon
    - Segmenting player engagement by loyalty/rank
    - Analyzing item pricing strategies and reward participation
- To derive actionable insights that can guide product balancing, marketing strategies, and player retention techniques.

Implementation of Relation Model via MySQL

As shown in the screenshots below, the relational model has been implemented in MySQL and MySQL Workbench is used to query the database. Sample data has been populated in all datasets and sample queries have been presented along with the sample output.

Snapshot of the database:



```
#select * from pokemon_battle_item
#select * from pokemon_held_item
#select * from player_event_participates
#select * from username
#select * from store
#select * from outfit
#select * from license
#select * from accessory
#select * from pokemon_move
#select * from player
#select * from matches
#select * from build_use
#select * from battle_records
```

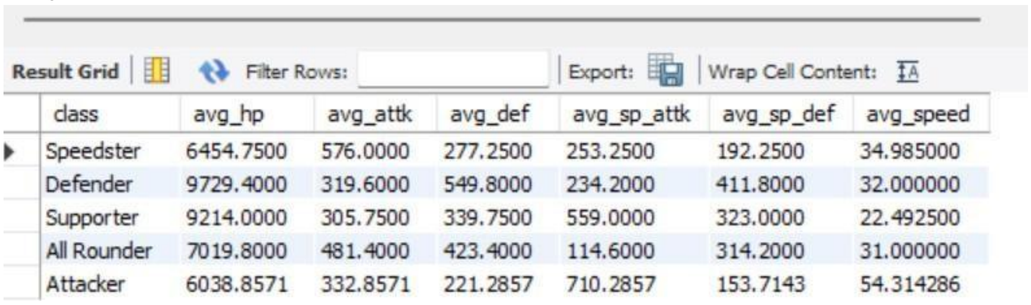
Battle_ID	Match_ID	Player_ID	Build_ID	Result
1	1	1	4	Win
2	1	12	8	Win
3	1	14	12	Win
4	1	17	16	Win
5	1	5	20	Win
6	1	6	8	Loss
7	1	3	24	Loss
8	1	2	32	Loss
9	1	11	12	Loss

Query 1

Analytical Purpose: To get an understanding of the average stats of Pokémon by the class. This can be used to adjust the stats of characters to balance the game when a new character is released

```
select class,
        avg(hp) as avg_hp,
        avg(attk) as avg_attk,
        avg(def) as avg_def,
        avg(sp_attk) as avg_sp_attk,
        avg(sp_def) as avg_sp_def,
        avg(speed) as avg_speed
from game.pokemon
group by class;
```

Output:



class	avg_hp	avg_attk	avg_def	avg_sp_attk	avg_sp_def	avg_speed
Speedster	6454.7500	576.0000	277.2500	253.2500	192.2500	34.985000
Defender	9729.4000	319.6000	549.8000	234.2000	411.8000	32.000000
Supporter	9214.0000	305.7500	339.7500	559.0000	323.0000	22.492500
All Rounder	7019.8000	481.4000	423.4000	114.6000	314.2000	31.000000
Attacker	6038.8571	332.8571	221.2857	710.2857	153.7143	54.314286

Query 2

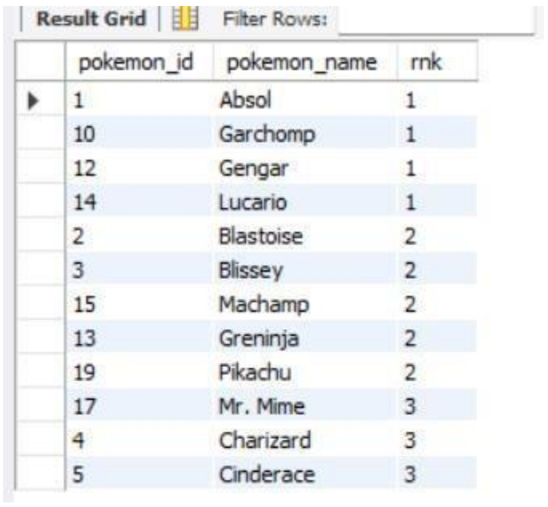
Analytical Purpose: Identifying the top 3 highest selling Pokémon in the shop to get an understanding of what kind of Pokémon are customers willing to spend money on.

```
with base as
(select product_id,
            count(*) as cnt
 from game.player_store_buys
 where store_name = "Unite License Committee"
 group by product_id)

select a.product_id as pokemon_id,
           b.name as pokemon_name,
           a.rnk
from (select product_id,
        dense_rank() over(order by cnt desc) as rnk
           from base) a
left join game.pokemon b
           on a.product_id = b.id

where a.rnk<=1;
```

Output



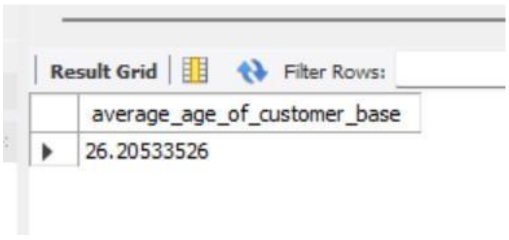
pokemon_id	pokemon_name	rnk
1	Absol	1
10	Garchomp	1
12	Gengar	1
14	Lucario	1
2	Blastoise	2
3	Blissey	2
15	Machamp	2
13	Greninja	2
19	Pikachu	2
17	Mr. Mime	3
4	Charizard	3
5	Cinderace	3

Query 3

Analytical Purpose: Determining the average age of the game's customer base

```
select avg(age) as average_age_of_customer_base
from (select email,
                datediff(CURDATE(), DOB)/365 as age
       from game.user) a;
```

Output:



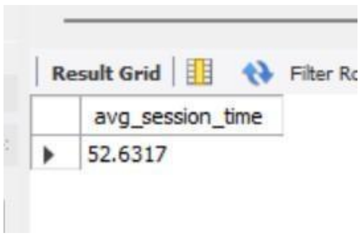
```
average_age_of_customer_base
26.20533526
```

Query 4

Analytical Purpose: Determining the average session time to get an understanding of how much time users spend per session. This will help us understand user activity and strategize what the user is exposed to in that interval.

```
### The time here is in minutes
select avg(TIMESTAMPDIFF(MINUTE, login_time, logout_time)) as avg_session from
game.activity_history;
```

Output:



```
avg_session_time
52.6317
```

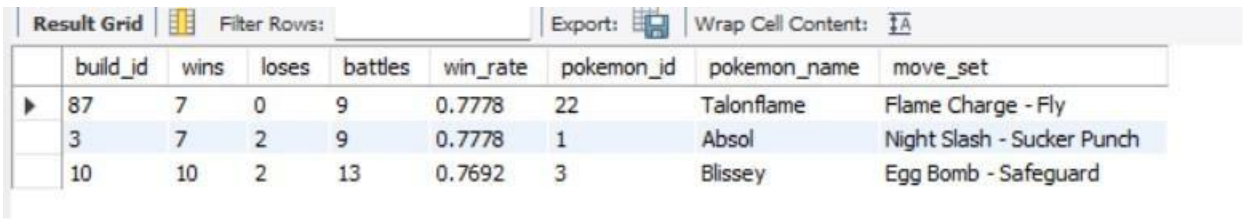
Query 5

Analytical Purpose: Determining the top 3 performing builds. By identifying this, the build can be made nerfed or modified to balance the game. Here, the build should be part of the minimum number of battles (threshold value of 6 in this sample) to judge the quality of the build accurately.

```
with    base    as
(select build_id,
          count(case when result="Win" then battle_id else null end) as wins,
          count(case when result="Lose" then battle_id else null end) as loses,
          count(battle_id)  as battles,
          count(case when result="Win" then battle_id else null end)/count(battle_id) as win_rate
 from game.battle_records
 group by build_id
 having battles>6)

select a.*,
         c.pokemon_id,
         d.name as pokemon_name,
         c.move_set
from base a
left join game.build_use c
      on a.build_id = c.build_id
left join game.pokemon d
      on c.pokemon_id = d.id
where 3 > (select count(*)
                 from base b
                 where a.win_rate < b.win_rate)
order by win_rate desc;
```

Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

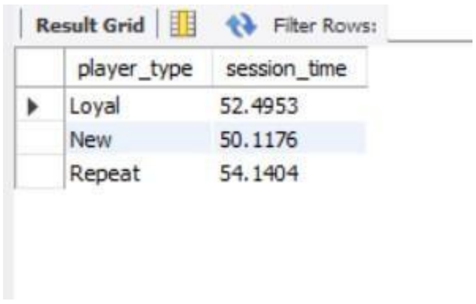
build_id	wins	loses	battles	win_rate	pokemon_id	pokemon_name	move_set
87	7	0	9	0.7778	22	Talonflame	Flame Charge - Fly
3	7	2	9	0.7778	1	Absol	Night Slash - Sucker Punch
10	10	2	13	0.7692	3	Blissey	Egg Bomb - Safeguard

Query 6

Analytical Purpose: Analysing the average time spent on the game by type of players from different ranks.

```
select case when b.rank < 10 then 'New'
                 when b.rank between 11 and 20 then 'Repeat'
                 when b.rank > 20 then 'Loyal'
           end as player_type,
           avg(TIMESTAMPDIFF(MINUTE, login_time, logout_time)) as session_time
from activity_history a
left join player b
               on a.player_id = b.ID
group by case when b.rank <= 10 then 'New'
                     when b.rank between 11 and 20 then 'Repeat'
                     when b.rank > 20 then 'Loyal'
               end;
```

Output:



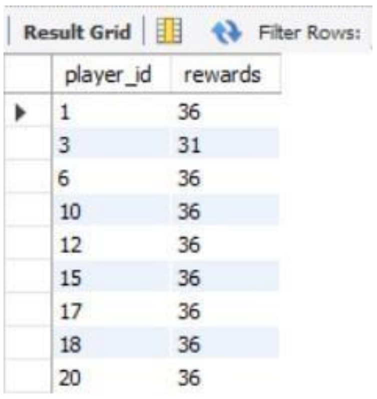
player_type	session_time
Loyal	52.4953
New	50.1176
Repeat	54.1404

Query 7

Analytical Purpose: Getting the list of players who have collected over 30 (threshold value in the sample) rewards. This can be used to flag these players as the ones who participate in events and play regularly. They can be targeted to sell items through events

```
select player_id,
        count(reward_id) as rewards
from game.player_event_rewards
group by player_id
having rewards>=30;
```

Output:



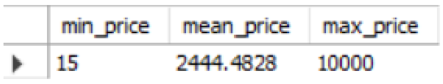
player_id	rewards
1	36
3	31
6	36
10	36
12	36
15	36
17	36
18	36
20	36

Query 8

Analytical Purpose: Getting a sense of the price range of items by calculating aggregate statistics in the item table

```
select min(price) as min_price,
        avg(price) as mean_price,
        max(price) as max_price
from game.item;
```

Output:



```
min_price   mean_price   max_price
15          2444.4828    10000
```