CPSC 304 Project Cover Page

Milestone #: 4

Date: 2022/11/21

Group Number: 29

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Gloria Mo	99556797	h5h3b	gloriamo321@gmail.com
Akash Raut	51475432	n4e0i	asraut29@gmail.com
Adrienne Chu	98338668	n9c3b	cadrienn368@gmail.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

2.
The SQL script is here (/project/src/SQLDDL.sql)
https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_h5h3b_n4e0i_n9c3b/blob/main/project/src/SQLDDL.sql

It should run as long as none of the tables already exist. If those tables already exist, you would need to uncomment the “DROP TABLE” lines at the top.

3.
Our repo is here:
https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_h5h3b_n4e0i_n9c3b

3.a)
Our final project is a database for hockey teams. There are tables with data for all the entities and relationships. There are functions for queries that insert, delete, update, and retrieve data for the Teams table (join also involves the Play table). There is also a GUI that can be used to trigger these functions. More detail for each query is in part 3.d and 3.e.

3.b)
The FDs in Teams and Tickets were removed. This is because they were arbitrarily made for Milestone 2 and didn’t really make a lot of sense. Removing them also simplified the design.

3.c)
Schema:

Underline = Primary Key/Candidate Key (we only have one candidate key each)
Bold = Foreign Key

Teams(home_city: string, name: string, net_worth: decimal, matches_played: int, wins: int, losses: int)

Team_Members(**home_city: string**, **team-name: string**, sin: int, member-name, age)

Coaches(**sin: int**)

Players(**sin: int,** jersey no.: int, role: string)

Play(**home_city: string**, **name: string**, **match_id: int**)

Train(**coach-sin: int**, **player-sin: int**)

Referees(id: int, name: string)

Oversee(**ref_id: int**, **match_id: int**)

Tickets(ticket_id: int, **sin.: int**, price: decimal, seat_no.: int, date_of_purchase: string, date_of_game: string)

Spectators(sin: int, name: string)

Watch(**match_id: int**, **sin: int**)

Matches(match_id: int, **tournament-name: string**, **season: int**, **stadium-name: string**, **location: string**, date_and_time: string)

Tournaments(name: string, season: int)

Stadiums(name: string, location: string)

Screenshots of tables after script is run:
Teams:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

Team_Members:

	HOME_CITY	TEAM_NAME	SIN	MEMBER_NAME	AGE
1	Vancouver	Canucks	123456789	Some Coach	44
2	Toronto	Maple Leafs	121212121	Sol Dudeguy	28
3	Montreal	Canadians	884974349	Big Bob	51
4	Edmonton	Oilers	1	Yoda	100
5	Calgary	Flames	339483239	Gigachad	69
6	Vancouver	Canucks	547932345	Some Player	21
7	Toronto	Maple Leafs	987654321	Ky Kooskey	23
8	Montreal	Canadians	7008009	Mudrock	29
9	Edmonton	Oilers	400400400	Luke	17
10	Calgary	Flames	483250283	Noob	18

Coaches:

	HOME_CITY	TEAM_NAME	SIN
1	Calgary	Flames	339483239
2	Edmonton	Oilers	1
3	Montreal	Canadians	884974349
4	Toronto	Maple Leafs	121212121
5	Vancouver	Canucks	123456789

Players:

	HOME_CITY	TEAM_NAME	SIN	JERSEY_NO	ROLE
1	Vancouver	Canucks	547932345	8	Forward
2	Toronto	Maple Leafs	987654321	7	Defence
3	Montreal	Canadians	7008009	1	Goalie
4	Edmonton	Oilers	400400400	4	Forward
5	Calgary	Flames	483250283	10	Defence

Train:

	COACH_HOME_CITY	COACH_TEAM_NAME	COACH_SIN	PLAYER_HOME_CITY	PLAYER_TEAM_NAME	PLAYER_SIN
1	Calgary	Flames	339483239	Calgary	Flames	483250283
2	Edmonton	Oilers	1	Edmonton	Oilers	400400400
3	Montreal	Canadians	884974349	Montreal	Canadians	7008009
4	Toronto	Maple Leafs	121212121	Toronto	Maple Leafs	987654321
5	Vancouver	Canucks	123456789	Vancouver	Canucks	547932345

Referees:

	ID	NAME
1	1	Broski
2	2	Buddy
3	3	Pal
4	4	Brotha
5	5	My Man

Spectators:

	SIN	NAME
1	196596859	Simp
2	295060560	#1 Fan
3	309450503	Rager
4	980850440	Mr. Watcher
5	294454095	John

Tournaments:

	NAME	SEASON
1	Cyberpunk	2077
2	Little League Hockey	2024
3	NHL League	2022
4	Piston Cup	2001
5	Stanley Cup	1998

Stadiums:

	NAME	LOCATION
1	Arasaka Tower	Night City
2	Death Star	Space
3	Pokemon League Building	Sinnoh
4	Rogers Arena	Vancouver
5	Some Place	Some City

Tickets:

	TICKET_ID	SIN	PRICE	DATE_OF_PURCHASE	SEAT_NO	DATE_OF_GAME
1	123	196596859	20	2022-04-20	1	2022-04-31
2	456	295060560	20	2010-01-19	2	2010-02-14
3	789	309450503	32	1988-12-31	3	1989-01-24
4	7	980850440	32	2044-03-03	4	2044-03-04
5	420	294454095	50	1999-11-14	5	1999-11-22

Matches:

	MATCH_ID	TOURNAMENT_NAME	SEASON	STADIUM_NAME	LOCATION	DATE_AND_TIME
1	1	Piston Cup	2001	Rogers Arena	Vancouver	2022-04-31 16:00
2	2	NHL League	2022	Some Place	Some City	2010-02-14 15:00
3	3	Little League Hockey	2024	Pokemon League Building	Sinnoh	1989-01-24 08:00
4	4	Stanley Cup	1998	Death Star	Space	2044-03-04 18:00
5	5	Cyberpunk	2077	Arasaka Tower	Night City	1999-11-22 21:00

Oversee:

	REF_ID	MATCH_ID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

Watch:

	MATCH_ID	SIN
1	1	196596859
2	1	980850440
3	2	196596859
4	2	295060560
5	2	980850440
6	3	196596859
7	3	309450503
8	3	980850440
9	4	196596859
10	4	980850440
11	5	196596859
12	5	294454095
13	5	980850440

Play:

	HOME_CITY	NAME	MATCH_ID
1	Calgary	Flames	3
2	Calgary	Flames	5
3	Edmonton	Oilers	2
4	Edmonton	Oilers	4
5	Montreal	Canadians	3
6	Montreal	Canadians	4
7	Toronto	Maple Leafs	1
8	Toronto	Maple Leafs	2
9	Vancouver	Canucks	1
10	Vancouver	Canucks	5

3.d), 3.e) a list of all sql queries used AND screenshots of the sample of the queries using the gui
Combining 3.d) and 3.e), the screenshots of the sample queries are shown after each of the SQL queries. The code is here: (/project/src/database/DatabseConnectionHandler.java): https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_h5h3b_n4e0i_n9c3b/blob/main/project/src/database/DatabaseConnectionHandler.java

Insert:

Insertion is shown in the **insertTeam()** function. The user must provide all of the attributes of the new Team. Then, it inserts a new entry with the given data into the Teams table.

Before insert:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The insert input:

After insert (new entry at bottom):

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11
16	NewCity	NewName	1	0	2	2

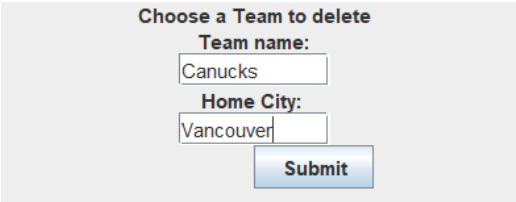
Delete:

Deletion is shown in the **deleteTeam()** function. The user must provide the key of the team to be deleted (home_city, name). Then, the team with that key is deleted from the table.

Before delete:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The delete input:



After delete (Vancouver Canucks is gone from the top):

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Toronto	Maple Leafs	200000	5	5	10
2	Montreal	Canadians	100000	10	0	10
3	Edmonton	Oilers	5000	7	4	11
4	Calgary	Flames	700000	1	12	13
5	Vancouver	Rocket	9001	8	11	19
6	Toronto	Flare	69000	20	29	49
7	Montreal	Plasma	77778	12	8	20
8	Edmonton	Black Knights	5030	7	14	21
9	Vancouver	Britannia	42000	18	14	32
10	Toronto	Evil Strike Force	200300	27	5	32
11	Montreal	Galactic	1	10	8	18
12	Edmonton	Magma	7800	13	14	27
13	Calgary	Aqua	700933	8	13	21
14	Calgary	Skull	78730	0	11	11

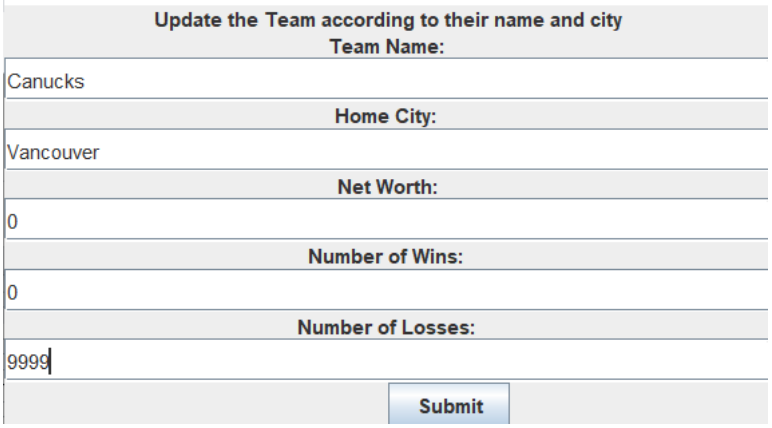
Update:

Updating is shown in the **updateTeams()** function. The user must provide the key for an existing team, and values for all the attributes of that team. Then, the team with the given key will be updated to have all of the provided new values.

Before update:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The update input:



After update (Vancouver Canucks is updated at the top):

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	0	0	9999	9999
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

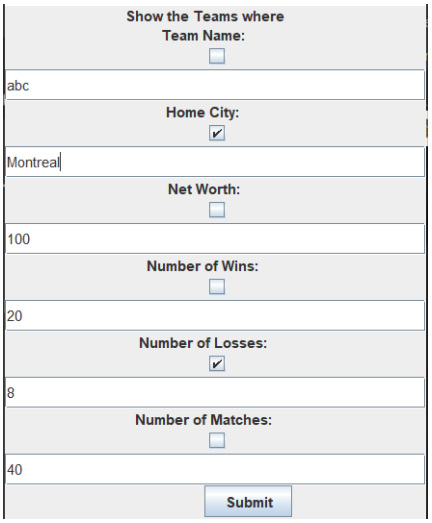
Selection:

Selection is shown in the **selectionTeams()** function. The user must input the attributes of Teams that they want to select based on, and also input what they want those attributes to be equal to (this selection function only allows the condition to be =). Then, they will receive an output of the entries in Teams with the specified conditions.

The Teams table:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The selection input:

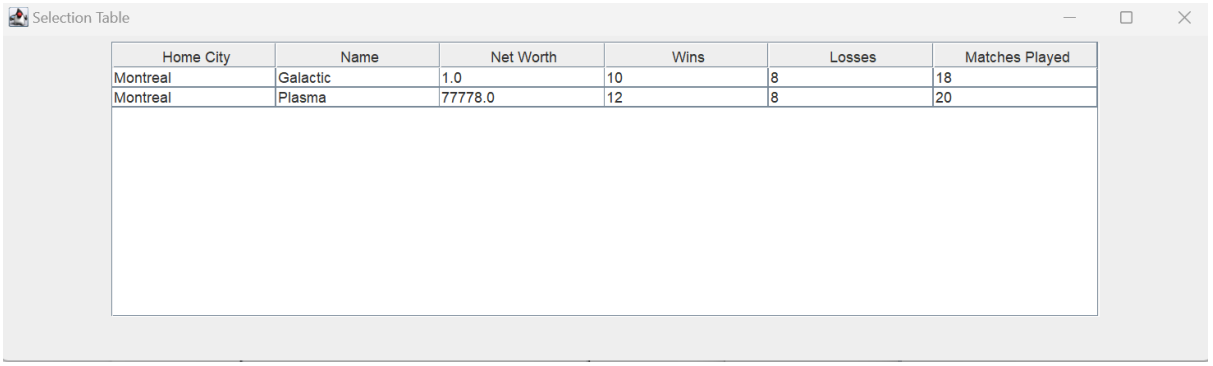


The output of selecting for selecting teams in Montreal with 8 losses:

Console Output:

```
Montreal Plasma 77778.0 12 8 20
Montreal Galactic 1.0 10 8 18
```

Gui Output:


Selection Table
Home City
Name
Net Worth
Wins
Losses
Matches Played
Montreal
Galactic
1.0
10
8
18
Montreal
Plasma
77778.0
12
8
20

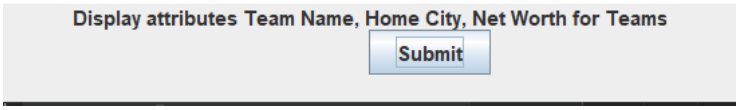
Projection:

Projection is shown in the **projectionTeams()** function. It is hard coded to project the home_city, name, and net_worth columns from the Teams table.

The Teams table:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The projection input:

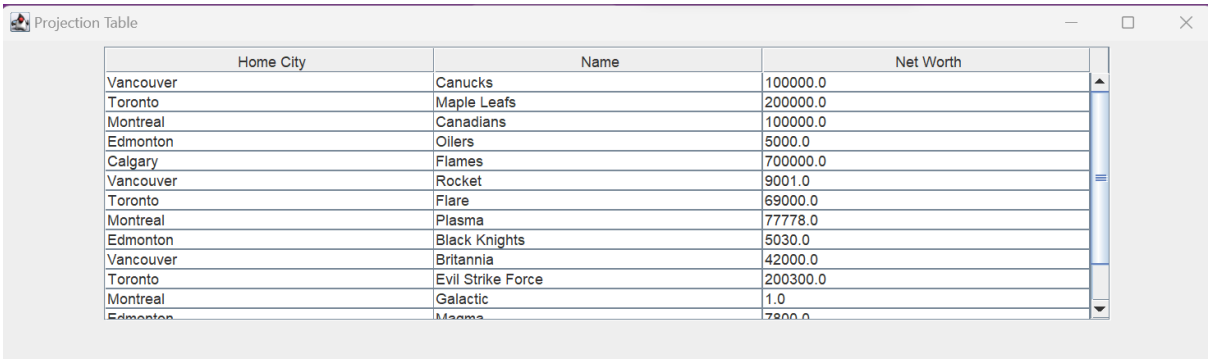


The output of the projection:

Console Output:

```
Vancouver Canucks 100000.0
Toronto Maple Leafs 200000.0
Montreal Canadians 100000.0
Edmonton Oilers 5000.0
Calgary Flames 700000.0
Vancouver Rocket 9001.0
Toronto Flare 69000.0
Montreal Plasma 77778.0
Edmonton Black Knights 5030.0
Vancouver Britannia 42000.0
Toronto Evil Strike Force 200300.0
Montreal Galactic 1.0
Edmonton Magma 7800.0
Calgary Aqua 700933.0
Calgary Skull 78730.0
```

Gui Output:



Home City	Name	Net Worth
Vancouver	Canucks	100000.0
Toronto	Maple Leafs	200000.0
Montreal	Canadians	100000.0
Edmonton	Oilers	5000.0
Calgary	Flames	700000.0
Vancouver	Rocket	9001.0
Toronto	Flare	69000.0
Montreal	Plasma	77778.0
Edmonton	Black Knights	5030.0
Vancouver	Britannia	42000.0
Toronto	Evil Strike Force	200300.0
Montreal	Galactic	1.0
Edmonton	Magma	7800.0

Join:

Joining is shown in the **joinTeamsPlays()** function. It joins Teams and Play. The user inputs a match_id, and the function joins Play and Team where the play's match_id matches the user's inputs. Then, it outputs the resulting teams' home_city, name, wins, and losses.

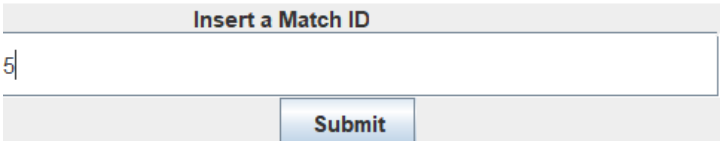
The Teams table:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The Play table:

	HOME_CITY	NAME	MATCH_ID
1	Calgary	Flames	3
2	Calgary	Flames	5
3	Edmonton	Oilers	2
4	Edmonton	Oilers	4
5	Montreal	Canadians	3
6	Montreal	Canadians	4
7	Toronto	Maple Leafs	1
8	Toronto	Maple Leafs	2
9	Vancouver	Canucks	1
10	Vancouver	Canucks	5

The join input:

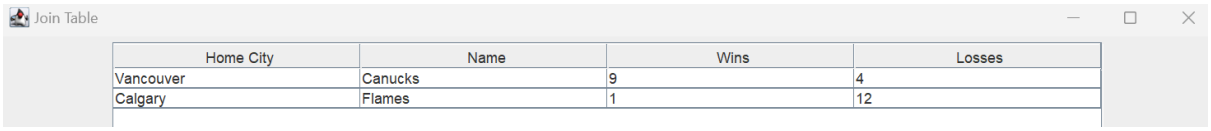


The output of the join with match_id = 5:

Console Output

```
Vancouver Canucks 9 4
Calgary Flames 1 12
```

Gui Output:



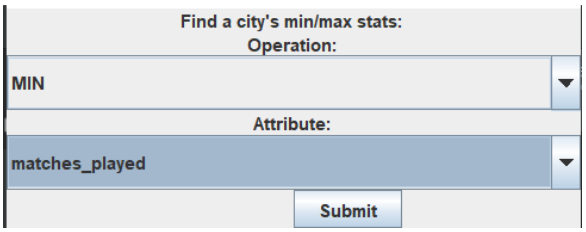
Aggregation with Group By:

Aggregation with group by is shown in the **aggregationGBTeams()** function. It takes a user input of an operator (such as min or max) and an attribute from Teams. Then, it returns each city with the value of the given attribute that matches the given operator.

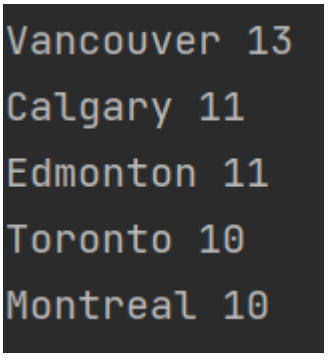
The Teams table:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

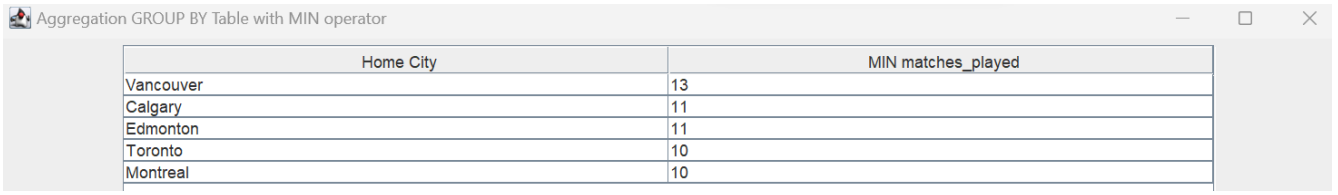
The input:



The output, where the operator was “min” and the attribute was “matches_played”:
Console:



Gui:



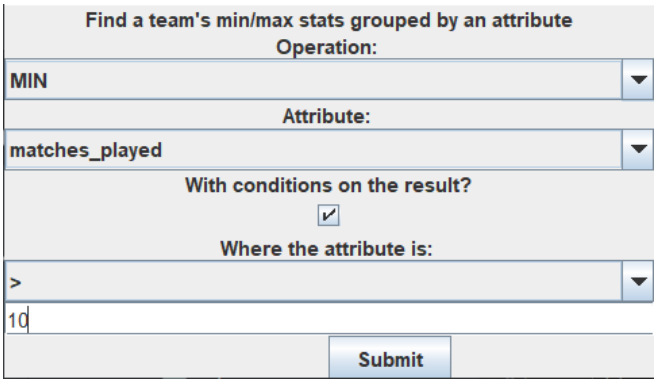
Aggregation with Having:

Aggregation with having is shown in the **aggregationHavingTeams()** function. It takes a user input of an operator (such as min or max) and an attribute from Teams. It also takes a condition to filter the output by. Then, it returns each city with the value of the given attribute that matches the given operator and condition.

The Teams table:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The input:

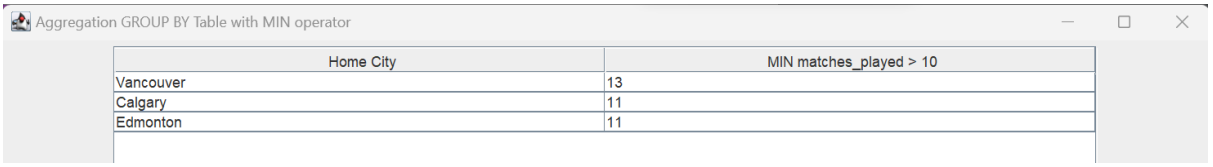


The output, where the operation was “min”, the attribute was “matches_played”, and there was a condition saying the attribute had to be > 10:

Console:

```
Vancouver 13
Calgary 11
Edmonton 11
```

Gui:



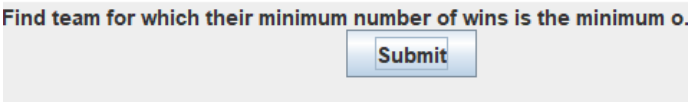
Nested Aggregation with Group By:

Nested aggregation with group by is shown in the **aggregateNested()** function. It is hard coded to output the home_city with the lowest amount of wins in Teams.

The Teams table:

	HOME_CITY	NAME	NET_WORTH	WINS	LOSSES	MATCHES_PLAYED
1	Vancouver	Canucks	100000	9	4	13
2	Toronto	Maple Leafs	200000	5	5	10
3	Montreal	Canadians	100000	10	0	10
4	Edmonton	Oilers	5000	7	4	11
5	Calgary	Flames	700000	1	12	13
6	Vancouver	Rocket	9001	8	11	19
7	Toronto	Flare	69000	20	29	49
8	Montreal	Plasma	77778	12	8	20
9	Edmonton	Black Knights	5030	7	14	21
10	Vancouver	Britannia	42000	18	14	32
11	Toronto	Evil Strike Force	200300	27	5	32
12	Montreal	Galactic	1	10	8	18
13	Edmonton	Magma	7800	13	14	27
14	Calgary	Aqua	700933	8	13	21
15	Calgary	Skull	78730	0	11	11

The input:

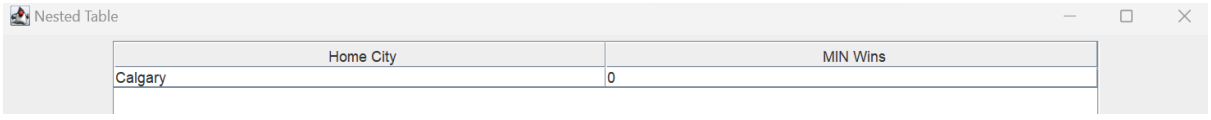


The output:

Console:

```
Calgary 0
```

Gui:



Division:

Division is shown in the **divisionSpectators()** function. The function finds all spectators who have watched all matches.

The Spectators table:

	SIN	NAME
1	196596859	Simp
2	295060560	#1 Fan
3	309450503	Rager
4	980850440	Mr. Watcher
5	294454095	John

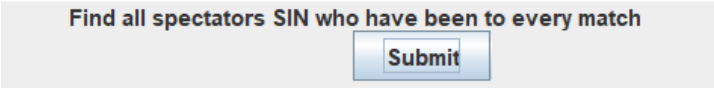
The Watch table:

	MATCH_ID	SIN
1	1	196596859
2	1	980850440
3	2	196596859
4	2	295060560
5	2	980850440
6	3	196596859
7	3	309450503
8	3	980850440
9	4	196596859
10	4	980850440
11	5	196596859
12	5	294454095
13	5	980850440

The Matches table:

	MATCH_ID	TOURNAMENT_NAME	SEASON	STADIUM_NAME	LOCATION	DATE_AND_TIME
1	1	Piston Cup	2001	Rogers Arena	Vancouver	2022-04-31 16:00
2	2	NHL League	2022	Some Place	Some City	2010-02-14 15:00
3	3	Little League Hockey	2024	Pokemon League Building	Sinnoh	1989-01-24 08:00
4	4	Stanley Cup	1998	Death Star	Space	2044-03-04 18:00
5	5	Cyberpunk	2077	Arasaka Tower	Night City	1999-11-22 21:00

The input:

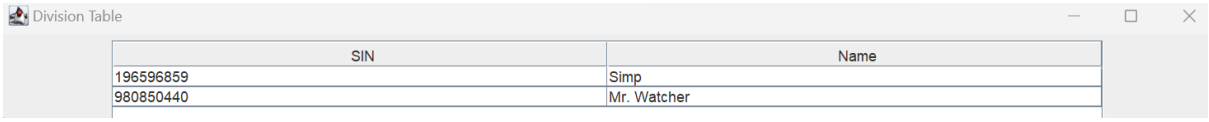


The output of the function:

Console:

```
196596859 Simp
980850440 Mr. Watcher
```

Gui:



4.
Our README is here:
https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_h5h3b_n4e0i_n9c3b/blob/main/README.md