University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4-5

Date: November 25th, 2022

Group Number: 67

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Michelle Nguyen	99641524	y6u4x	mxnguyyen@gmail.com
Jordan So	29669314	y0t4q	jordanso7789@gmail.com
Jack Parkinson	13956917	g5h3b	jackparkinson1394@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

a. Overview / Purpose of the Project:

Our project is based around a volleyball database that contains information on players, teams, leagues, countries and more that pertain to volleyball leagues. The purpose of this project is to construct this database and create a graphical interface to alter and display elements of the database. For example, we provide an option on the user interface to update a country name and thus update all tuples containing that country name. This project could be used to model and alter data on real volleyball leagues through oracle, which would be much more efficient then having to write down all the information on pen and paper.

b. Differences from Previous Schema:

There are three major differences:

- **The Player relation was refactored into the ISA child relations** (libero, setter, etc). We decided to do this because in the original schema there was redundant information in the player table and the child table because the ISA relation was total disjoint. As a result, player relation was refactored into the child position relations.
- **'PlayerBMI' was broken down into 5 different position BMI tables.** As a result of the Player relation refactoring discussed above, the PlayerBMI was broken down into individual position BMI tables (i.e. LiberoBMI, SetterBMI, etc.). 5 different relations had to be made because the same Player ID can exist in multiple position tables.
- **Play's relation was refactored into the game relation**. Because the Game to Team relationship can be represented as many to one if you include the winning and losing team ID in every Game tuple, then the original play relation was deleted.

There are also many minor changes such as:

- League Table- CountryName now NOT NULL
- HeadCoach Table - YearsCoaching is now FK to Experience Table, added TID which is FK to Team
- CoachExperience Table- CID is now FK to HeadCoach
- StadiumAddress Table- SID is now FK to Stadium
- Team Table - LID now NOT NULL
- Game Table - LID now NOT NULL, removed SID as a PK
- Players Table - TID now NOT NULL

- ER Diagram at end of document

c. Copy of Schema + Screenshots of data:

Country(Cname: Varchar(20), Population:Integer)

```
SQL> select * from country
  2  ;

CNAME                POPULATION
-------------------- ----------
Tobi                     300000
Canada                 38010000
USA                   329500000
Italy                  59550000
Brazil                212600000
```

League(LID: Integer, **Cname**: Varchar(20), Name: Char(40))

```
SQL> select * from league;

       LID CNAME                NAME
---------- -------------------- ----------------------------------------
         1 Canada               The Canadian League
         2 USA                  American Volleyball Open
         3 Italy                Superlega
         4 Brazil               Super League
```

Stadium(SID: Integer, Name: Char(40))

```
SQL> select * from stadium;

       SID NAME
---------- ----------------------------------------
         1 Stadium One
         2 Sydney Cricket Ground
         3 Maracana Stadium
         4 Staples Center
         5 Accor Arena
```

StadiumAddress(**SID**: Integer, City: Char(40), Address: Char(40))

```
SQL> select * from stadiumAddress
  2  ;

       SID CITY
---------- ----------------------------------------
ADDRESS
----------------------------------------
         1 Vancouver
1234 Numbers Street

         2 Sydney Cricket Ground
Driver Ave

         3 Maracana Stadium
Maracana Road


       SID CITY
---------- ----------------------------------------
ADDRESS
----------------------------------------
         4 Staples Center
1111 S Figueroa Street

         5 Accor Arena
Bercy Boulevard
```

StadiumName(City: Char(20), Address: Char(40), Name: Char(40))

```
SQL> select * from stadiumName
  2  ;

CITY                 ADDRESS
-------------------- ----------------------------------------
NAME
----------------------------------------
Vancouver            1234 Numbers Street
Stadium One

Sydney               Driver Ave
Sydney Cricket Ground

Rio de Janeiro       Maracana Road
Maracana Stadium


CITY                 ADDRESS
-------------------- ----------------------------------------
NAME
----------------------------------------
Los Angeles          1111 S Figueroa Street
Staples Center

Paris                Bercy Boulevard
Accor Arena
```

Ref(RID: Integer, Name: Char(20), Salary: Integer)

```
SQL> select * from ref;

       RID NAME                      SALARY
---------- -------------------- ----------
         1 Charlotte Leclerc         45000
         2 Sam Sedin                 67000
         3 Michael Jordan           400000
         4 Eric Demko                23000
         5 Pam Fleming               50000
```

Experience(YearsCoaching: Integer, Experience: Char(20))

```
SQL> select * from experience;

YEARSCOACHING EXPERIENCE
------------- --------------------
            4 Beginner
            8 Veteran
           17 Veteran
            1 Beginner
            5 Intermediate
```

HeadCoach(CID: Integer, Name: Char(20), **YearsCoaching**: Integer)

```
SQL> select * from headCoach;

       CID NAME                 YEARSCOACHING
---------- -------------------- -------------
         1 Jason Jenkins                   4
         2 Mary Smith                      8
         3 Geoffrey Green                 17
         4 Fletcher Flin                   1
         5 Tiffany Thompson                5
```

CoachExperience(CID: Integer, **Experience**: Char(20))

```
SQL> select * from coachExperience
  2  ;

       CID EXPERIENCE
---------- --------------------
         1 Beginner
         2 Veteran
         3 Veteran
         4 Beginner
         5 Intermediate
```

Table(TID: Integer, TeamName: Char(20), Record: Integer, **LID**: Integer, CID: Integer)

```
SQL> select * from team;

       TID TEAMNAME                RECORD        LID        CID
---------- -------------------- ---------- ---------- ----------
         1 Dolphins                      5          1          2
         2 Volley ViKings                9          1          4
         3 Dolphins                      3          1          2
         4 Templeton Titans              4          2          5
         5 Apex14U                       1          2          1
```

Game(GID: Integer, **SID**: Integer, **WinTID**: Integer, **LoseTID**: Integer, **LID**: Integer)

```
SQL> select * from game;

       GID        SID     WINTID    LOSETID        LID
---------- ---------- ---------- ---------- ----------
         1          1          1          2          1
         2          1          3          1          1
         3          1          2          3          2
         4          1          1          2          1
         5          2          4          5          2
```

GameSet(SetNumber: Integer, **GID**: Integer, WinnerScore: Integer, LoserScore: Integer)

```
SQL> select * from gameSet;

 SETNUMBER        GID WINNERSCORE LOSERSCORE
---------- ---------- ----------- ----------
         1          1          25         20
         2          1          25         20
         1          2          15         19
         2          2          15         16
         3          2          15         13
```

WorksFor(**RID**: Integer, **LID**: Integer)

```
SQL> select * from worksFor
  2  ;

       RID        LID
---------- ----------
         1          1
         1          2
         1          3
         1          4
         1          5
         2          5
         3          2
         4          2
```

RefsFor(**RID**: Integer, **GID**: Integer)

```
SQL> select *  from refsFor
  2  ;

       RID        GID
---------- ----------
         1          1
         2          2
         3          3
         4          4
         5          5
```

Libero(PID: Integer, Name: Char(20), Weight: Integer, Height: Integer, Digs: Integer, JerseyNumber: Integer, **TID**: Integer)

```
SQL> select * from libero;

       PID NAME                 WEIGHT     HEIGHT       DIGS JERSEYNUMBER
---------- -------------------- ---------- ---------- ---------- ------------
       TID
----------
         1 Bob Smith                    60        160         10           14
         1

         2 Steve Mathhews               65        170         20           10
         2

         3 Frank Manger                 60        170         15            1
         3


       PID NAME                 WEIGHT     HEIGHT       DIGS JERSEYNUMBER
---------- -------------------- ---------- ---------- ---------- ------------
       TID
----------
         4 Brook Pol                    50        150         30            8
         4

         5 Mor West                     55        180         22            2
         5
```

ServerSpecialist(PID: Integer, Name: Char(20), Weight: Integer, Height: Integer, Aces: Integer, JerseyNumber: Integer, **TID**: Integer)

```
SQL> select * from serverSpecialist;

       PID NAME                    WEIGHT     HEIGHT       ACES JERSEYNUMBER
---------- -------------------- ---------- ---------- ---------- ------------
       TID
----------
         1 Jarvis Gibbs                 60        160         18           14
         1

         2 Jonah Muir                   65        170         28           10
         2

         3 Waqar Herrera                60        170         22           11
         3


       PID NAME                    WEIGHT     HEIGHT       ACES JERSEYNUMBER
---------- -------------------- ---------- ---------- ---------- ------------
       TID
----------
         4 Reagan Castro                50        150         15           18
         4

         5 Gianni Snider                55        180         21           12
         5
```

OutsideHitter(PID: Integer, Name: Char(20), Weight: Integer, Height: Integer, Kills: Integer, Aces: Integer, Blocks: Integer, JerseyNumber: Integer, **TID**: Integer)

```
SQL> select * from outsideHitter;

       PID NAME                    WEIGHT     HEIGHT      KILLS       ACES
---------- -------------------- ---------- ---------- ---------- ----------
    BLOCKS JERSEYNUMBER        TID
---------- ------------ ----------
         1 Elliot Metcalfe              60        160         20         10
        15           21          1

         2 Arlo Conrad                  65        170         18         15
        10           20          2

         3 Adyan Herman                 60        170         14         16
        18           25          3


       PID NAME                    WEIGHT     HEIGHT      KILLS       ACES
---------- -------------------- ---------- ---------- ---------- ----------
    BLOCKS JERSEYNUMBER        TID
---------- ------------ ----------
         4 Jarred Jensen                50        150         12         14
        13           28          4

         5 Stan Dougherty               55        180         21         11
        19           24          5
```

Setter(PID: Integer, Name: Char(20), Weight: Integer, Height: Integer, SetAttempts: Integer, SetSuccessRate: Decimal(2,2), JerseyNumber: Integer, **TID**: Integer)

```
SQL> SELECT * FROM Setter;

       PID NAME                    WEIGHT     HEIGHT SETATTEMPTS SETSUCCESSRATE
---------- -------------------- ---------- ---------- ----------- --------------
JERSEYNUMBER        TID
------------ ----------
         1 Steffan Connelly            60        160          92            .78
          32          1

         2 Remy Davidson               65        170          80            .82
          31          2

         3 Damian Mclaughlin           60        170         102            .91
          34          3


       PID NAME                    WEIGHT     HEIGHT SETATTEMPTS SETSUCCESSRATE
---------- -------------------- ---------- ---------- ----------- --------------
JERSEYNUMBER        TID
------------ ----------
         4 Jeanne Mullins              50        150          40            .67
          35          4

         5 Amir Ridley                 55        180          74            .71
          39          5
```

MiddleBlocker(PID: Integer, Name: Char(20), Weight: Integer, Height: Integer, Blocks: Integer, JerseyNumber: Integer, **TID**: Integer)

```
SQL> SELECT * FROM MiddleBlocker;

       PID NAME                    WEIGHT     HEIGHT     BLOCKS JERSEYNUMBER
---------- -------------------- ---------- ---------- ---------- ------------
       TID
----------
         1 Hashim Hodges               60        160         22           32
         1

         2 Vijay Spence                65        170         21           31
         2

         3 Dominic Levine              60        170         12           34
         3


       PID NAME                    WEIGHT     HEIGHT     BLOCKS JERSEYNUMBER
---------- -------------------- ---------- ---------- ---------- ------------
       TID
----------
         4 Blane Davis                 50        150         35           35
         4

         5 Sal Goodman                 55        180         31           39
         5
```

BMI(Height: Integer, Weight: Integer, BMI: Decimal(3,1))

```
SQL> select * from bmi;

    HEIGHT     WEIGHT        BMI
---------- ---------- ----------
       160         60       23.4
       170         65       22.5
       170         60       20.8
       150         50       22.2
       180         55         17
```

LiberoBMI(**PID**: Integer, BMI: Decimal(3,1))

```
SQL> select * from liberoBMI
  2  ;

       PID        BMI
---------- ----------
         1       23.4
         2       22.5
         3       20.8
         4       22.2
         5         17
```

ServerSpecialistBMI(**PID**: Integer, BMI: Decimal(3,1))

```
SQL> select * from serverSpecialistBMI;

       PID        BMI
---------- ----------
         1       23.4
         2       22.5
         3       20.8
         4       22.2
         5         17
```

OutsideHitterBMI(**PID**: Integer, BMI: Decimal(3,1))

```
SQL> select * from outsideHitterBMI;

       PID        BMI
---------- ----------
         1       23.4
         2       22.5
         3       20.8
         4       22.2
         5         17
```

SetterBMI(**PID**: Integer, BMI: Decimal(3,1))

```
SQL> SELECT * FROM SetterBMI;

       PID         BMI
---------- ----------
         1        23.4
         2        22.5
         3        20.8
         4        22.2
         5          17
```

MiddleBlockerBMI(**PID**: Integer, BMI: Decimal(3,1))

```
SQL> SELECT * FROM MiddleBlockerBMI;

       PID         BMI
---------- ----------
         1        23.4
         2        22.5
         3        20.8
         4        22.2
         5          17
```

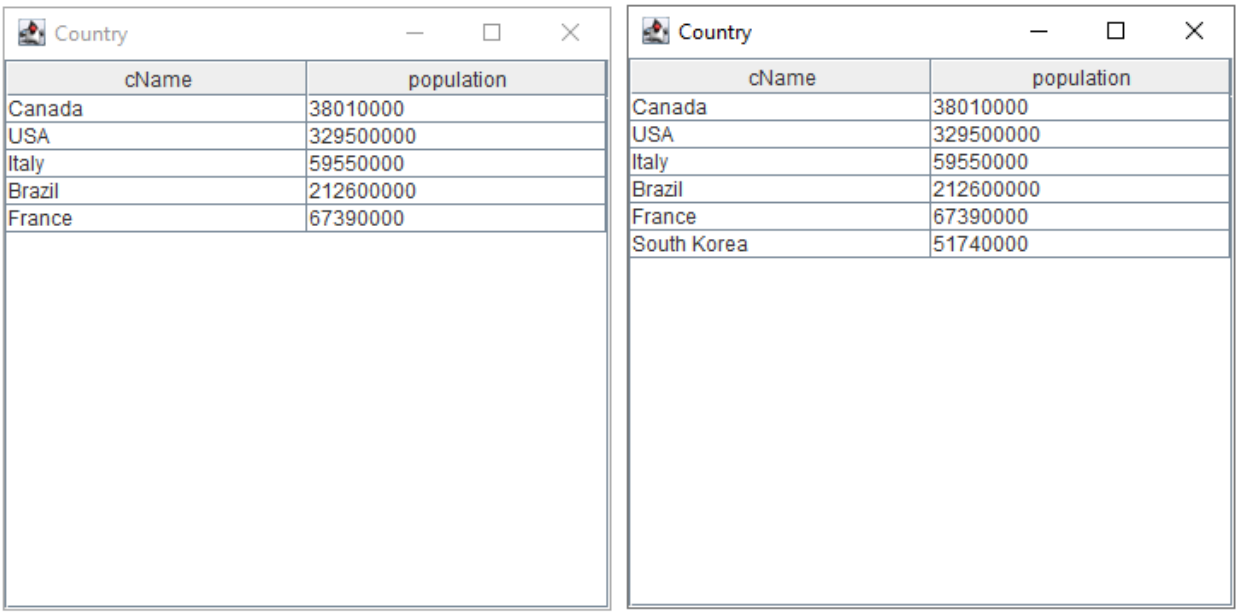
d. List of SQL queries:

- Inserting new tuples into Country
- Deleting tuples in Country
- Updating tuples in Country
- Selection on all relations
- Projection on all relations
- Join on all relations
- Division: Find referees that ref in all leagues
- Aggregation Having: Show all game ids, GIDS, of games with 3 sets
- Aggregation Group By: Find the average salary of refs in each league
- Nested Aggregation: Show all refs that work for more than one league

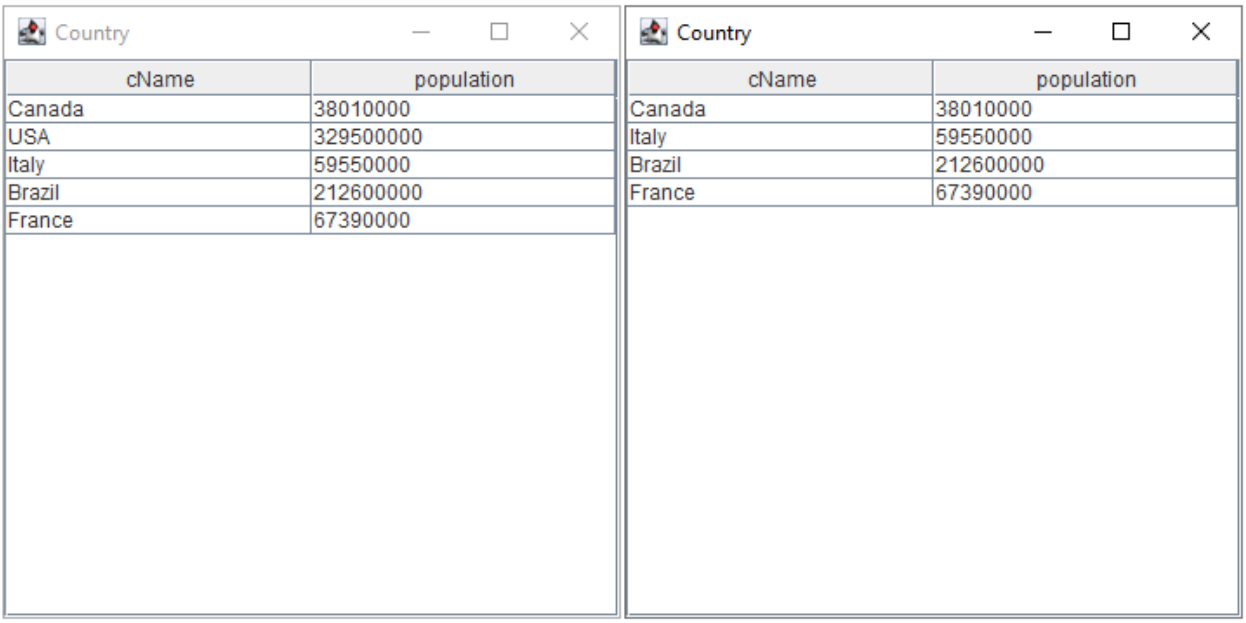
e. Screenshots of output of Queries w/ GUI

Implementations for all queries can be found in DatabaseConnectionHandler.java in database package. GUI portion of the implementation is found in the ui package where there is a class for each query.

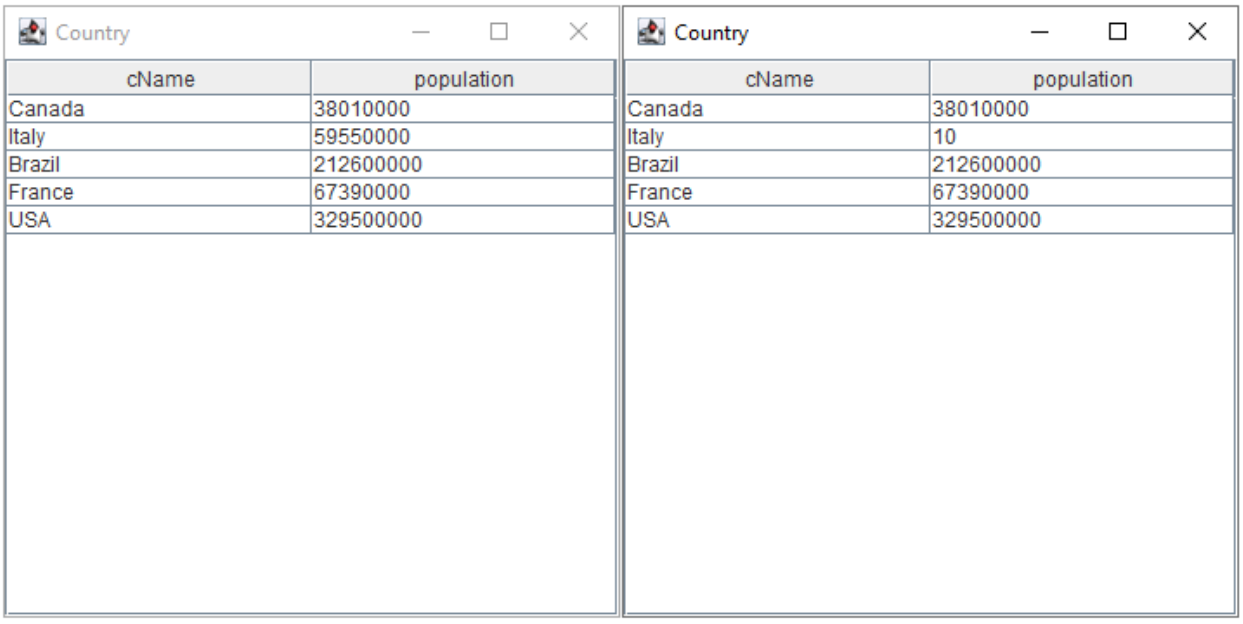
Insert Query:
INSERT INTO Country VALUES (‘South Korea’,’51740000’)
Reference: insertCountry(Country model) in DatabaseConnectionHandler.java
Before and after:



Delete Query:
DELETE FROM Country WHERE Cname=’USA’
Reference: deleteCountry(String countryName) in DatabaseConnectionHandler.java
Before and after:

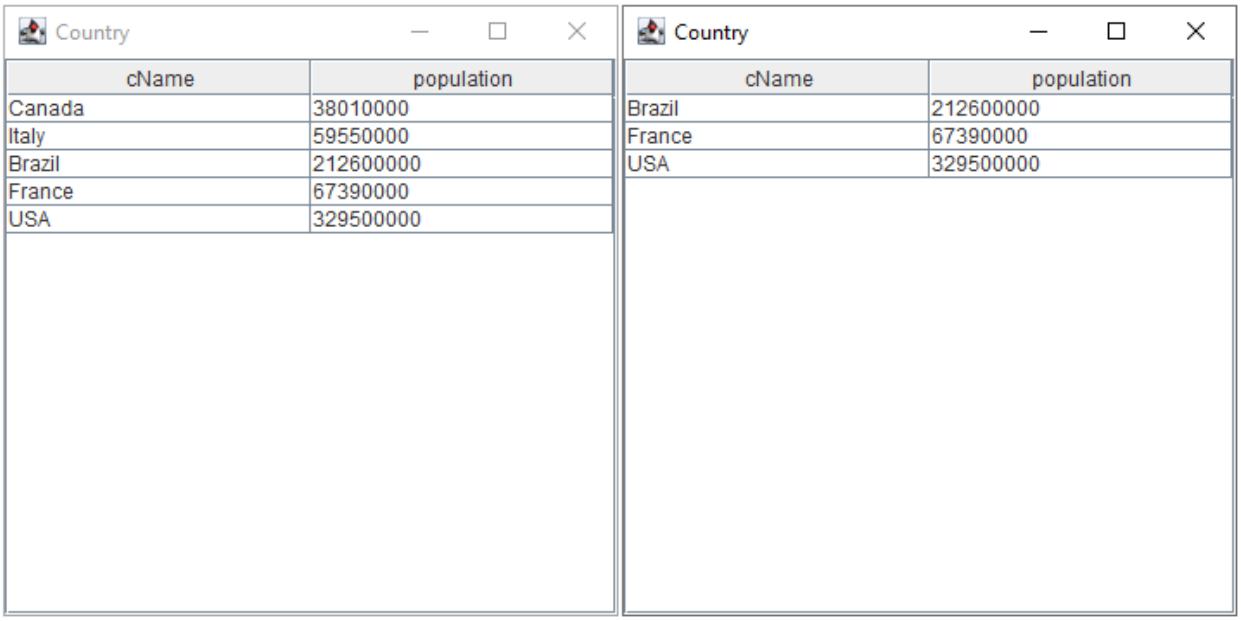


Update Query:
UPDATE Country SET Population='10' WHERE Cname='Italy'
Reference: updateCountry(String countryName, int population) in DatabaseConnectionHandler.java
Before and after:



Selection Query:
SELECT * FROM Country WHERE population>'60000000'
Reference: getRelationInfo(Relation relation, String[] conditions) in DatabaseConnectionHandler.java

Before and after:



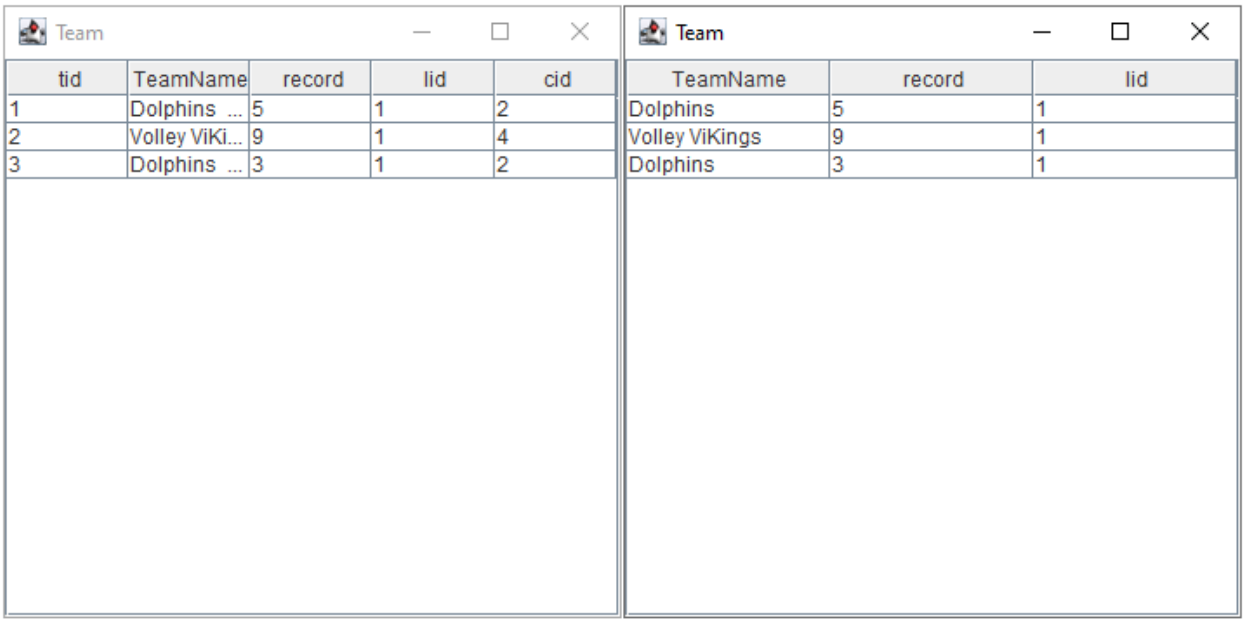
Country

cName	population
Canada	38010000
Italy	59550000
Brazil	212600000
France	67390000
USA	329500000

Country

cName	population
Brazil	212600000
France	67390000
USA	329500000

Projection Query:
SELECT TeamName, record, lid FROM Team
Reference: getProjectionInfo(Relation relation, String[] attributes) in DatabaseConnectionHandler.java
Before and after:



Team

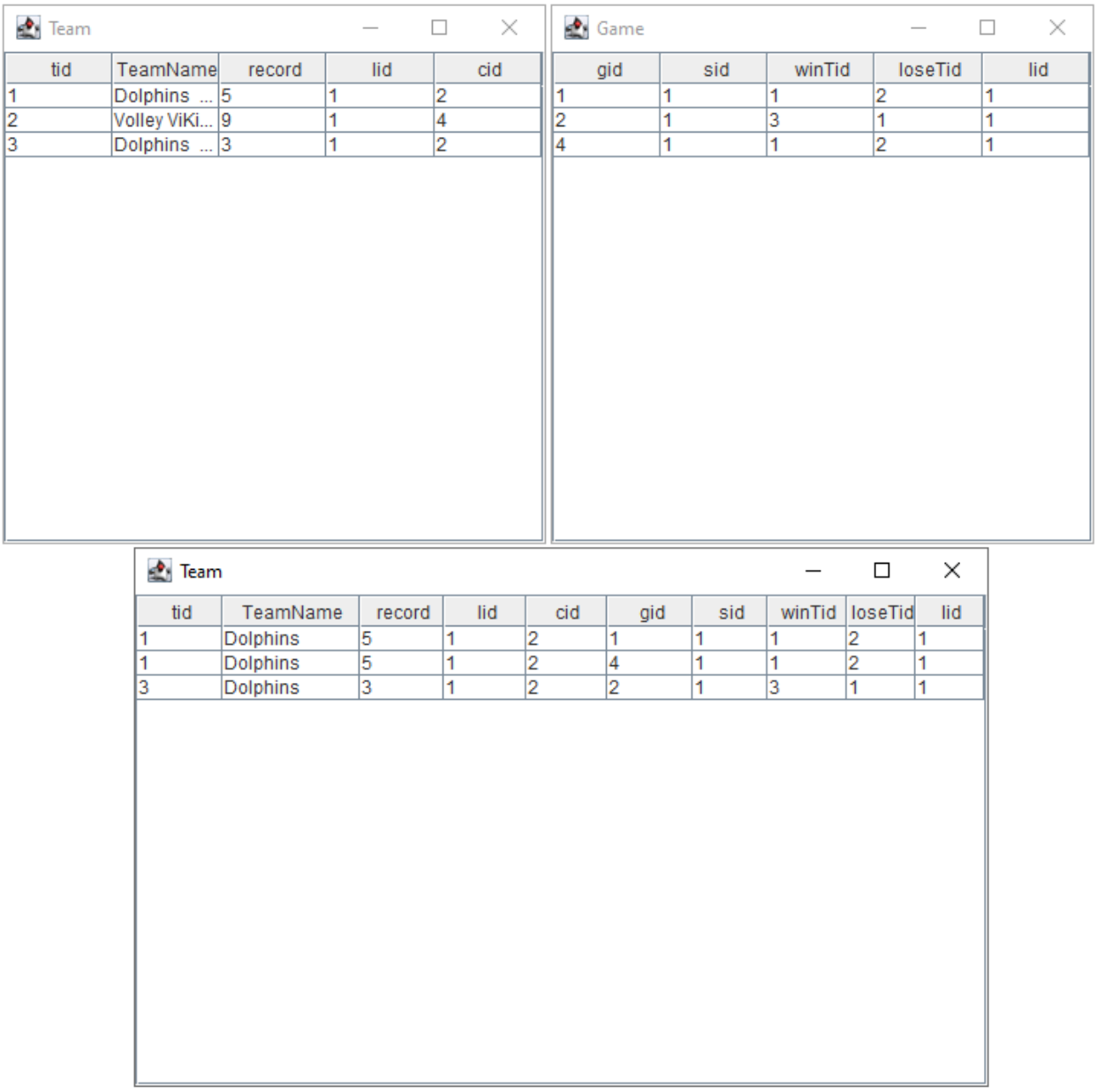
tid	TeamName	record	lid	cid
1	Dolphins ...	5	1	2
2	Volley ViKi...	9	1	4
3	Dolphins ...	3	1	2

Team

TeamName	record	lid
Dolphins	5	1
Volley ViKings	9	1
Dolphins	3	1

Join Query:
SELECT * FROM Team t, Game g WHERE t.tid = g.winTid
Reference: getJoinInfo() in DatabaseConnectionHandler.java

Before and after:



Team

tid	TeamName	record	lid	cid
1	Dolphins ...	5	1	2
2	Volley ViKi...	9	1	4
3	Dolphins ...	3	1	2

Game

gid	sid	winTid	loseTid	lid
1	1	1	2	1
2	1	3	1	1
4	1	1	2	1

Team

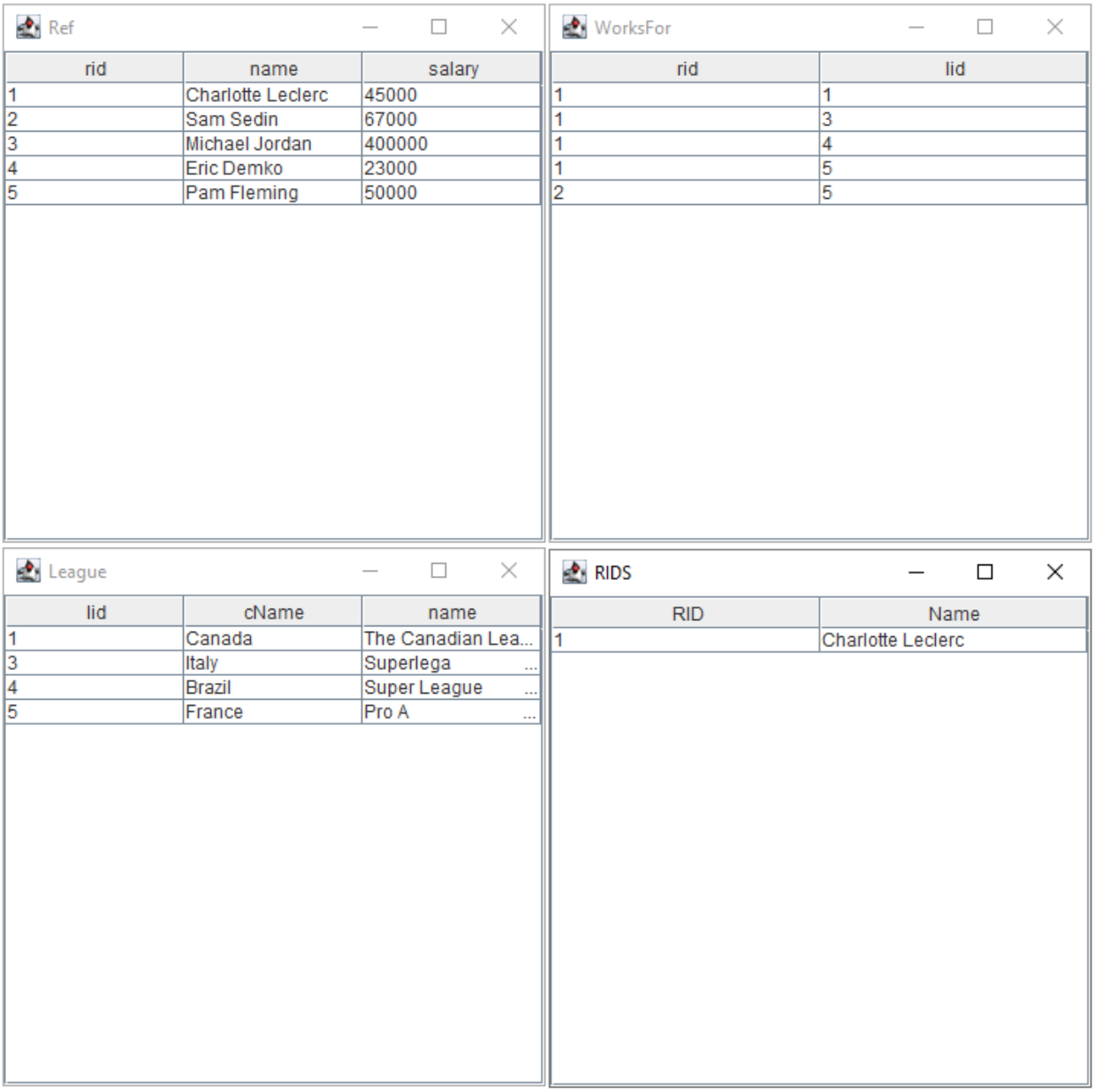
tid	TeamName	record	lid	cid	gid	sid	winTid	loseTid	lid
1	Dolphins	5	1	2	1	1	1	2	1
1	Dolphins	5	1	2	4	1	1	2	1
3	Dolphins	3	1	2	2	1	3	1	1

Division Query:

```
SELECT RID, Name FROM Ref R WHERE NOT EXISTS
        (SELECT L.lid FROM League L WHERE NOT EXISTS
                (SELECT W.rid FROM WorksFor W WHERE L.lid = W.rid AND W.rid = R.rid))
```

Reference: getDivisionRef() in DatabaseConnectionHandler.java

Before and after: RIDS is output table



Ref

rid	name	salary
1	Charlotte Leclerc	45000
2	Sam Sedin	67000
3	Michael Jordan	400000
4	Eric Demko	23000
5	Pam Fleming	50000

WorksFor

rid	lid
1	1
1	3
1	4
1	5
2	5

League

lid	cName	name
1	Canada	The Canadian Lea...
3	Italy	Superlega ...
4	Brazil	Super League ...
5	France	Pro A ...

RIDS

RID	Name
1	Charlotte Leclerc

Aggregation Group By Query:
SELECT LID, AVG(Salary) as avgSalary
FROM Ref r, WorksFor w
WHERE r.RID = w.RID
GROUP BY LID
Reference: getAggregationGroup() in DatabaseConnectionHandler.java
Before and after:

Ref

rid	name	salary
1	Charlotte Leclerc	45000
2	Sam Sedin	67000
3	Michael Jordan	400000
4	Eric Demko	23000
5	Pam Fleming	50000

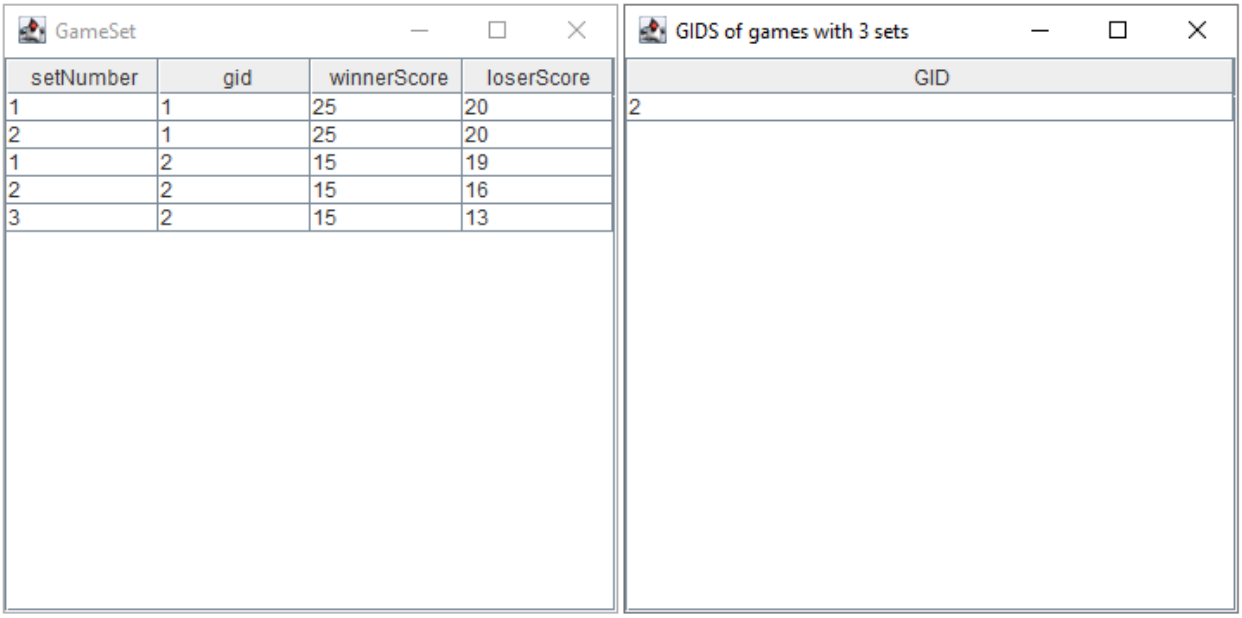
WorksFor

rid	lid
1	1
1	3
1	4
1	5
2	5

Average Salary of refs in each lea...

Leauge ID	Average Salary
1	45000
4	45000
5	56000
3	45000

Aggregation Having Query:
SELECT GID From GameSet
GROUP BY GID
HAVING COUNT(*)=3
Reference: getAggregationHaving() in DatabaseConnectionHandler.java
Before and after:



Nested Aggregation Query:

```
SELECT r.RID, r.Name
FROM Ref r
WHERE 1 < (SELECT COUNT(*)
                FROM WorksFor w
                WHERE w.rid = r.rid
                GROUP BY w.rid)
```

Reference: getAggregationNested() in DatabaseConnectionHandler.java

Before and after:

Ref

rid	name	salary
1	Charlotte Leclerc	45000
2	Sam Sedin	67000
3	Michael Jordan	400000
4	Eric Demko	23000
5	Pam Fleming	50000

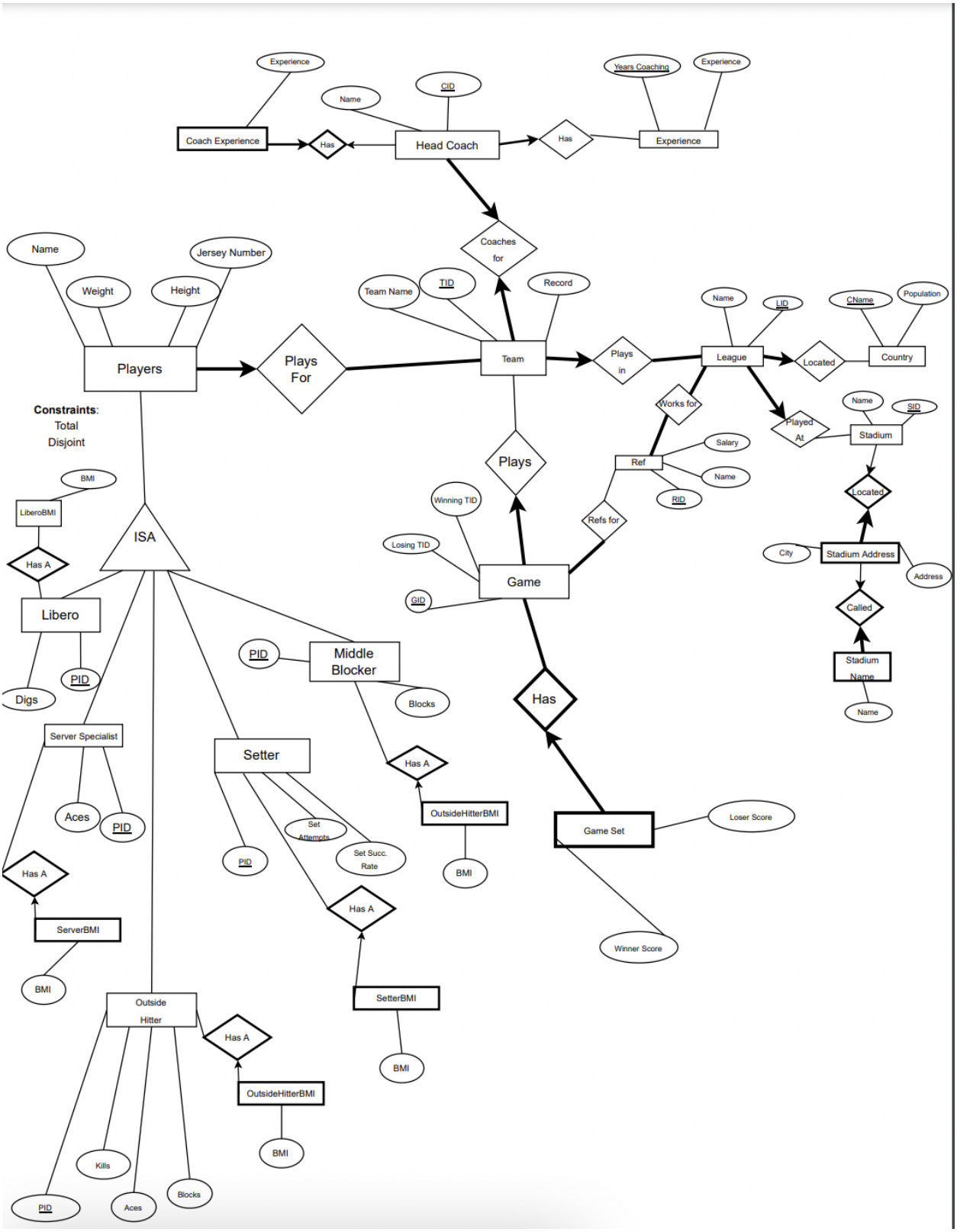
WorksFor

rid	lid
1	1
1	3
1	4
1	5
2	5

Refs that work for more than one ...

RID	Name
1	Charlotte Leclerc

ER Diagram:


Experience
Name
CID
Years Coaching
Experience
Coach Experience
Has
Head Coach
Has
Experience
Coaches for
Name
Jersey Number
Weight
Height
Team Name
TID
Record
Name
LID
CName
Population
Players
Plays For
Team
Plays in
League
Located
Country
Constraints:
Total
Disjoint
Works for
Name
SID
Played At
Stadium
Plays
Ref
Salary
Name
RID
BMI
LiberoBMI
ISA
Winning TID
Refs for
Located
Losing TID
Game
City
Stadium Address
Address
Has A
Libero
GID
Called
PID
Middle Blocker
Stadium Name
Digs
PID
Blocks
Has
Name
Server Specialist
Setter
Has A
Aces
PID
Set Attempts
Set Succ. Rate
OutsideHitterBMI
Game Set
Loser Score
PID
BMI
Has A
Has A
ServerBMI
SetterBMI
Winner Score
BMI
BMI
Outside Hitter
Has A
OutsideHitterBMI
Kills
BMI
PID
Aces
Blocks