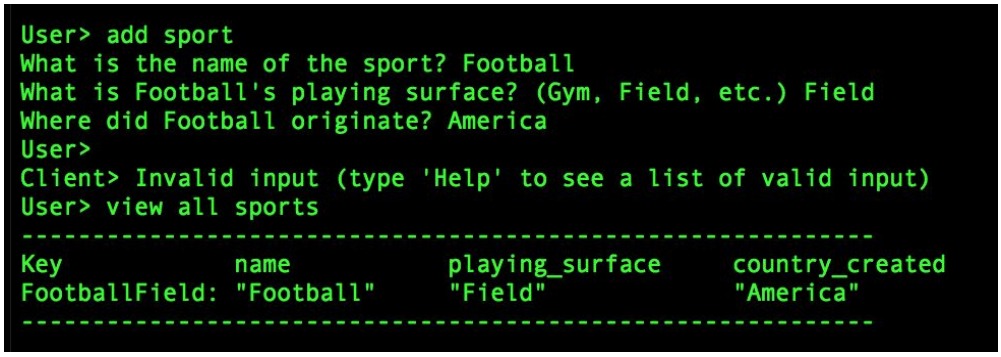
This first image shows what the user sees on Client startup. He is greeted with a lovely message and a nicely formatted menu of available commands.

Here, we see the user has decided to add a new sport to his database. We also see what happens when the user tries to pass in an empty string (invalid input). We also see the user request to see all sports, of which there is only one - the one he has just added.



Next, the user adds a team. He makes sure to set the team's sport to one of the available ones. He then requests to see all teams, and is shown two (one of them was created in a previous session).

```
User> add team
What is the name of the team? Aggies
Which Sport does Aggies play? Football
How many wins do the Aggies have? 6
How many losses do the Aggies have? 0
How many ties do the Aggies have? 0
User> view all teams
------------------------------------------------------------
Key      name           sport          wins    losses  ties
Aggies6: "Aggies"       "Football"     6       0       0
Bucks0:  "Bucks"        "Football"     0       6       0
------------------------------------------------------------
```

Now the user decides to add a player. After giving in the data, he views the team the new player belongs to and sees the new player listed at the bottom with all the proper data.

```
User> add player
What is the new player's name? Cory
What team does Cory play for? Aggies
How old is Cory? 21
What is Cory's Jersey Number? 08
What is Cory's position? Linebacker
User> view team
What is the name of the team you want to view? Aggies
------------------------------------------------------------
Key      name           age     jersey  position
Speedy1: "Speedy"       21      1       "Runner"
loser99: "loser"        18      99      "None"
Cory08:  "Cory"         21      08      "Linebacker"
------------------------------------------------------------
```

Next, the player decides to update a team's values with the new win count after the Tennessee game last week. Notice how the user types casually which values he wants to update.

```
User> update team
What is the name of the team to update? Aggies
Which Sport does Aggies play? Football
How many wins do the Aggies have? 6
Which values do you want to update? (Pick as many as you like: Name, Sport, Wins, Losses,
 or Ties) Wins, and maybe losses too
How many wins do the Aggies have now? 100
How many losses do the Aggies have now? 0
User> view all teams
------------------------------------------------------------
Key        name           sport          wins    losses  ties
Aggies100: "Aggies"       "Football"     100     0       0
Bucks0:    "Bucks"        "Football"     0       6       0
------------------------------------------------------------
```

Next, the user decides to update a player's information. Cory has turned 22, so that is the value he updates. The change is immediately reflected in the table shown.

```
User> update player
What is the name of the player to update? Cory
What team does Cory play for? Aggies
What is Cory's Jersey Number? 08
Which values do you want to update? (Pick as many as you like: Name, Age, Team, Jersey, o
r Position) Just age
How old is Cory's now? 22
User> view team
What is the name of the team you want to view? Aggies
-------------------------------------------------------
Key      name           age     jersey  position
Speedy1: "Speedy"       21      1       "Runner"
loser99: "loser"        18      99      "None"
Cory08:  "Cory"         22      08      "Linebacker"
-------------------------------------------------------
```

Actually, it might be best to remove Cory from the database. He's just been kicked off the team, and the user likes to have his database as accurate as possible. After giving a few specific attribute values, the Client successfully removes Cory from the database.

```
User> remove player
What is the name of the player to delete? Cory
What team does Cory play for? Aggies
What is Cory's Jersey Number? 08
User> view team
What is the name of the team you want to view? Aggies
-------------------------------------------------------
Key      name           age     jersey  position
Speedy1: "Speedy"       21      1       "Runner"
loser99: "loser"        18      99      "None"
-------------------------------------------------------
```

Here, the user adds 2 new sports to the database

```
User> add sport
What is the name of the sport? Basketball
What is Basketball's playing surface? (Gym, Field, etc.) Gym
Where did Basketball originate? America
User> add sport
What is the name of the sport? Rock Climbing
What is Rock_Climbing's playing surface? (Gym, Field, etc.) Cliff
Where did Rock_Climbing originate? Britain
```

Here we see the results of the previous additions

```
User> view all sports
-------------------------------------------------------
Key                 name           playing_surface     country_created
FootballField:      "Football"     "Field"             "America"
BasketballGym:      "Basketball"   "Gym"               "America"
Rock_ClimbingCliff: "Rock_Climbing""Cliff"             "Britain"
-------------------------------------------------------
```

Here, the user simply wanted to know exactly which teams were in the database. A quick command will show all stored teams.

```
User> view all teams
-------------------------------------------------------------
Key         name            sport          wins    losses  ties
Aggies100:  "Aggies"        "Football"     100     0       0
Bucks0:     "Bucks"         "Football"     0       6       0
-------------------------------------------------------------
```

Here, we see the same thing with players.

```
User> view all players
-----------------------------------------------------------
Key       name            age     team            jersey  position
Speedy1:  "Speedy"        21      "Aggies"        1       "Runner"
Johnny2:  "Johnny"        23      "Aggies"        2       "Quarterback"
loser99:  "loser"         18      "Bucks"         99      "None"
-----------------------------------------------------------
```

Say the two separate teams “Aggies” and “Bucks” were suddenly merged together as the result of the two schools merging. The user merges the teams here and is shown the result.

```
User> merge teams
What is the name of the first team you want to merge? Aggies
What is the name of the second team you want to merge? Bucks
-----------------------------------------------------------
Key       name            age     jersey  position
Speedy1:  "Speedy"        21      1       "Runner"
loser99:  "loser"         18      99      "None"
Johnny2:  "Johnny"        23      2       "Quarterback"
-----------------------------------------------------------
```

Here, the user decides to trade a player between two teams. This is easily accomplished with the “trade player” command.

```
User> trade player
What is the name of the first player to trade? Speedy
What team does Speedy play for? Aggies
What is Speedy's Jersey Number? 1
What is the name of the second player to trade? Johnny
What team does Johnny play for? Bucks
What is Johnny's Jersey Number? 2
```

Here is the result of trading the players for the team “Aggies”

```
User> view team
What is the name of the team you want to view? Aggies
------------------------------------------------------------
Key      name           age     jersey  position
loser99: "loser"        18      99      "None"
Johnny2: "Johnny"       23      2       "Quarterback"
------------------------------------------------------------
```

And here’s the result of trading the players for the team “Bucks”

```
User> view team
What is the name of the team you want to view? Bucks
------------------------------------------------------------
Key      name           age     jersey  position
Speedy1: "Speedy"       21      1       "Runner"
------------------------------------------------------------
```

Finally, the user decides he is finished with his current session and quits the program.

```
User> quit
Disconnecting from server... Connection closed.
66-76-8-237:RDBMS coryavra$
```

On exit, the Server saves all relations that aren't views (as assigned by the homework guidelines)

```
Server stream:
------------------------------------------------------------
EXIT;
------------------------------------------------------------

Engine stream:
------------------------------------------------------------
Tokenized ArrayList: [EXIT, ;]
EXIT invoked
Serialized data is saved in table_data/Aggies.ser
Serialized data is saved in table_data/Bucks.ser
Error: is a view. Failed to write Temp Expression Table2.
Serialized data is saved in table_data/sports.ser
Serialized data is saved in table_data/teams.ser
Serialized data is saved in table_data/players.ser
Error: is a view. Failed to write Temp Natural Join Table.
Error: is a view. Failed to write Temp Expression Table.
Serialized data is saved in table_data/ComputersAreHard.ser
Error: is a view. Failed to write Temp Expression Table1.
Error: is a view. Failed to write Temp Set Union Table.
------------------------------------------------------------

Client successfully disconnected
66-76-8-237:RDBMS coryavra$
```

Here is some sample output on the Server side of things. This shows part of the "trade player" function.

```
Client stream:
------------------------------------------------------------
INSERT INTO Aggies VALUES FROM RELATION select (name=="Johnny"&&jersey==2) Bucks;
------------------------------------------------------------

Engine stream:
------------------------------------------------------------
Tokenized ArrayList: [INSERT, INTO, Aggies, VALUES, FROM, RELATION, select, (, name, ==,
"Johnny", &&, jersey, ==, 2, ), Bucks, ;]
INSERT invoked
SELECT invoked
Conditions List:[name, ==, "Johnny", &&, jersey, ==, 2]
Expressions List (or table name): [Bucks]
Inserted row with key Johnny2 in table Selection from Temp Expression Table
Error: Row doesn't exist. Cannot get.
Inserted row with key Johnny2 in table Aggies
------------------------------------------------------------
```

Here's an example of what happens when the user decides to quit mid-interrogation:

```
User> add sport
What is the name of the sport? Soccer
What is Soccer's playing surface? (Gym, Field, etc.) quit
Disconnecting from server... Connection closed.
```

Our program successfully cancels the interrogation and proceeds to safely close the program, saving all relations as it does so.

Here's an example of what happens when the user enters complete hogwash:

```
User> add player
What is the new player's name? dwank;l
What team does dwankl play for? dnklaw
How old is dwankl? knl
What is dwankl's Jersey Number? kldwam
What is dwankl's position? ;lk
```

Our program realizes that the data doesn't fit its specified domain, so it fails to insert (This is server-side)

```
Client stream:
------------------------------------------------------------
INSERT INTO players VALUES FROM ("dwankl", knl, "dnklaw", kldwam, "lk");
------------------------------------------------------------

Engine stream:
------------------------------------------------------------
Tokenized ArrayList: [INSERT, INTO, players, VALUES, FROM, (, "dwankl", knl, "dnklaw", kl
dwam, "lk", ), ;]
INSERT invoked
players["dwankl", knl, "dnklaw", kldwam, "lk"]
knl does not match its domain: INTEGER
Must be either 'VARCHAR(X)' or 'INTEGER'
kldwam does not match its domain: INTEGER
Must be either 'VARCHAR(X)' or 'INTEGER'
Error: Invalid value detected; failed to insert row.
------------------------------------------------------------
```