From: Chia Wen Cheng
To: SI 564 FA 23 Teaching Team
Re: SQL and Database Final Project-an introduction to the data exploration analysis of the NHL
Date: Dec 4, 2023

Dear teaching team,

I hope you all are taking good care of yourselves as the busiest season of the semester has approached.

Based on our previous face-to-face discussions, I've developed a database called "NHL" for your exploration of compelling data stories within the National Hockey League. The "NHL" database comprises 9 tables and over 27,000 rows of data, making it manageable for you to explore at your convenience. Please take your time, and don't feel overwhelmed! Additionally, due to time constraints, the database does not include data from before the year 1990. If you're keen on expanding the dataset to include earlier data, please feel free to inform me, and I'll gladly continue our collaboration.

The questions you raised in our meetings are outlined in Section I, while the answers, including queries and screenshots of results related to your intriguing questions, are provided in Section II. Given the diverse sources of data used in constructing this database, meticulous data cleaning was imperative to ensure the usability of the collected information. I will elaborate on my data collection, cleaning, and manipulation processes in Section III. For a visually clear Entity-Relationship Diagram (ERD), please refer to Section IV (the last page of this document excluding Appendix).

Section I.
Some questions you raised interest in knowing about are:

1. Players' salaries (people are always curious about professional sports players' salaries)
    a. What are the average salaries of each team?
    b. What are the average salaries of each drafted year and round? We expect players being drafted by an NHL team earlier in years and rounds to have higher average salaries. Is this hypothesis true?
    c. What are each position as well as handedness' expected salaries? (If you are thinking about changing your children's handedness to get higher pay as ice hockey players.)
2. What are the numbers of counts of players of each nationality?
3. Do Canadian hockey teams tend to have more Canadian players?
4. What is the maximum second stayed on ice grouped by player, team, year of game, and division?
5. How many times of championships do teams that have won the Stanley Cup win?
6. What is the saving percentage of the team in the year they win the championship?

Section II.
The answers to your questions including the queries for obtaining these answers are attached in this section.

Q1. What are the average salaries of each team?

In terms of average salaries, the Dallas Stars emerge as the most generous, with the New York Rangers, Los Angeles Kings, New York Islanders, Boston Bruins, and others following suit. On the contrary, the Atlanta Thrashers appear to have the lowest average salary, with a substantial gap between them and the next least-paid team. It's important to note that the Atlanta Thrashers are not currently an existing team, and this result may be influenced by currency changes. They relocated and became Winnipeg Jets. Luckily, the Winnipeg Jets are not getting the least paid! Conversely, the Minnesota Wild, Arizona Coyotes, Vancouver Canucks, San Jose Sharks, and Calgary Flames rank among the teams with the lowest overall average salary payments, which, are all currently affiliated with the Western Conference. 😏 This is certainly not a favorable indication if you're seeking a team for your children, or yourselves, to join!

```
mysql> select avg(s.salary) as average_salary, p.team_short from salary s join player p on s.player_id = p.player_id group by p.team_short orde
r by average_salary desc;
+----------------+------------+
| average_salary | team_short |
+----------------+------------+
|   2829090.1765 | DAL        |
|   2644449.2470 | NYR        |
|   2594701.3716 | LAK        |
|   2491615.4943 | NYI        |
|   2452539.8882 | BOS        |
|   2419214.9243 | STL        |
|   2361192.6923 | CHI        |
|   2324443.5207 | CAR        |
|   2302435.6897 | DET        |
|   2294222.9683 | SEA        |
|   2288138.5671 | VGK        |
|   2265785.9603 | FLA        |
|   2258614.7129 | TOR        |
|   2249326.8527 | MTL        |
|   2196355.0000 | PIT        |
|   2191823.7564 | EDM        |
|   2132450.5864 | NSH        |
|   2127162.3893 | TBL        |
|   2115896.0253 | WSH        |
|   1978404.3913 | CBJ        |
|   1978265.1901 | BUF        |
|   1978265.1901 | BUF        |
|   1974806.7829 | ANA        |
|   1955830.5950 | COL        |
|   1885364.4076 | NJD        |
|   1865406.5337 | PHI        |
|   1839516.2709 | WPG        |
|   1779985.3652 | OTT        |
|   1776602.9338 | CGY        |
|   1732546.3666 | SJS        |
|   1676348.4922 | VAN        |
|   1672261.4209 | ARI        |
|   1619188.7584 | MIN        |
|    650150.0000 | ATL        |
+----------------+------------+
33 rows in set (0.12 sec)
```

Q2. What are the average salaries of each drafted year and round? We expect players being drafted by an NHL team earlier in years and rounds to have higher average salaries. Is this hypothesis true?

Analyzing the displayed query results, we observe a trend where players drafted in earlier rounds tend to receive higher salaries. This outcome aligns with our expectations, as players drafted earlier are generally deemed more competitive among all candidates of that year, backed by professional statistics.

However, notably low average salaries are evident in the years 2015, 2013, 2012, and 2006. Potential explanations range from a single or a few data points disproportionately influencing the average amount for that year to external events that occurred, necessitating further research for a comprehensive understanding.

```
mysql> select avg(s.salary) as average_salary, p.year_drafted, p.round_drafted from salary s join player p on s.player_id = p.player_id group b
y p.year_drafted, p.round_drafted order by p.year_drafted desc;
+----------------+--------------+---------------+
| average_salary | year_drafted | round_drafted |
+----------------+--------------+---------------+
|   4416665.8654 |         2016 |             1 |
|   3958785.6061 |         2015 |             1 |
|   2308297.2222 |         2015 |             2 |
|    734000.0000 |         2015 |             3 |
|    711666.6667 |         2015 |             4 |
|    718214.2857 |         2015 |             5 |
|   1990714.2857 |         2015 |             7 |
|   3149958.1044 |         2014 |             1 |
|   2059142.8571 |         2014 |             2 |
|   3362142.8571 |         2014 |             3 |
|   1712583.3333 |         2014 |             4 |
|   1217625.0000 |         2014 |             5 |
|   2756562.5000 |         2014 |             6 |
|   1570000.0000 |         2014 |             7 |
|   3208527.9851 |         2013 |             1 |
|   1295575.0000 |         2013 |             2 |
|   2041061.6438 |         2013 |             3 |
|   1491869.4444 |         2013 |             4 |
|    732500.0000 |         2013 |             5 |
|    725104.1667 |         2013 |             6 |
|   1685000.0000 |         2013 |             7 |
|   1685000.0000 |         2013 |             7 |
|   2761956.0185 |         2012 |             1 |
|   1632106.7708 |         2012 |             2 |
|   2403869.3182 |         2012 |             3 |
|   1902465.7049 |         2012 |             4 |
|   1173750.0000 |         2012 |             5 |
|   1002309.2105 |         2012 |             6 |
|    709583.3333 |         2012 |             7 |
|   2978329.2429 |         2011 |             1 |
|   1910745.9677 |         2011 |             2 |
|   1383084.9057 |         2011 |             3 |
|   2179126.9841 |         2011 |             4 |
|   1295346.3542 |         2011 |             5 |
|   1421406.2500 |         2011 |             6 |
|   1623524.5902 |         2011 |             7 |
|   3489848.9711 |         2010 |             1 |
|   1977059.9174 |         2010 |             2 |
|   1401984.1270 |         2010 |             3 |
|   1490875.0000 |         2010 |             4 |
|   2005945.1220 |         2010 |             5 |
|   2096861.1111 |         2010 |             6 |
|   3669094.7540 |         2009 |             1 |
|   2426524.0000 |         2009 |             2 |
|   1853041.2371 |         2009 |             3 |
|   2004998.2000 |         2009 |             4 |
|   1653856.3830 |         2009 |             5 |
```

1653856.3830	2009	5
2063250.0000	2009	6
1228225.8065	2009	7
4238002.2831	2008	1
2250654.2969	2008	2
2278822.4783	2008	3
1743944.4444	2008	4
1334759.6154	2008	5
2306176.5873	2008	6
1611666.6667	2008	7
3760427.0833	2007	1
4627000.0000	2007	2
1576111.1111	2007	3
1405409.0909	2007	4
3313244.6809	2007	5
1889180.8511	2007	6
1857968.7500	2007	7
4548565.1515	2006	1
3109975.3521	2006	2
2079435.4839	2006	3
1561750.0000	2006	4
683958.3333	2006	5
2459054.0541	2006	6
1414166.6667	2006	7
4450312.5000	2005	1
3155416.6667	2005	2

3155416.6667	2005	2
3961764.7059	2005	3
2232887.0968	2005	4
1760357.1429	2005	5
6003369.5652	2004	1
4134183.6735	2004	2
2901266.6667	2004	3
2520740.7407	2004	4
3308333.2500	2004	5
1082142.8571	2004	6
2170000.0000	2004	7
1966470.5882	2004	9
5235000.0000	2003	1
5195384.6154	2003	2
3735000.0000	2003	3
3100000.0000	2003	4
1504629.6296	2003	5
1954464.2857	2003	6
3833333.3333	2003	7
6100000.0000	2003	8
2672058.8235	2003	9
4578787.8788	2002	1
4069642.8571	2002	2
3819565.2174	2002	3
3128571.4286	2002	4
4964285.7143	2002	8

4964285.7143	2002	8
3650000.0000	2002	9
4831707.3171	2001	1
2959843.7500	2001	2
4873529.4118	2001	3
1128125.0000	2001	4
4312500.0000	2001	5
2850000.0000	2001	6
3265625.0000	2001	7
3743396.2264	2000	1
3752173.9130	2000	2
1157142.8571	2000	3
2343750.0000	2000	5
1415000.0000	2000	6
6550000.0000	1999	1
2268750.0000	1999	3
1094444.4444	1999	5
4794736.8421	1999	7
3783333.3333	1998	2
3821875.0000	1998	3
3817857.1429	1998	6
5582608.6957	1997	1
1968750.0000	1997	5
6336750.0000	1997	6
2327777.7778	1996	2
5128750.0000	1996	3

4312500.0000	2001	5
2850000.0000	2001	6
3265625.0000	2001	7
3743396.2264	2000	1
3752173.9130	2000	2
1157142.8571	2000	3
2343750.0000	2000	5
1415000.0000	2000	6
6550000.0000	1999	1
2268750.0000	1999	3
1094444.4444	1999	5
4794736.8421	1999	7
3783333.3333	1998	2
3821875.0000	1998	3
3817857.1429	1998	6
5582608.6957	1997	1
1968750.0000	1997	5
6336750.0000	1997	6
2327777.7778	1996	2
5128750.0000	1996	3
5212581.0000	1995	1
1493301.7695	NULL	NULL

123 rows in set (0.16 sec)

For the purpose of comparing average salaries across different years, I narrow down the results to display only the average salaries of players drafted in the first round for the years under consideration. The findings indicate that players drafted earlier in the year tend to receive higher average salaries. This correlation may be attributed to the fact that players drafted earlier often have greater seniority until now, potentially leading to higher compensation. However, as I'm not a fervent sports analyst, I lack the expertise to assess the accuracy of this conjecture.

```
mysql> select avg(s.salary) as average_salary, p.year_drafted, p.round_drafted from salary s join player p on s.player_id = p.player_id where p
.round_drafted = 1 group by p.year_drafted, p.round_drafted order by p.year_drafted desc;
+----------------+--------------+---------------+
| average_salary | year_drafted | round_drafted |
+----------------+--------------+---------------+
|   4416665.8654 |         2016 |             1 |
|   3958785.6061 |         2015 |             1 |
|   3149958.1044 |         2014 |             1 |
|   3208527.9851 |         2013 |             1 |
|   2761956.0185 |         2012 |             1 |
|   2978329.2429 |         2011 |             1 |
|   3489848.9711 |         2010 |             1 |
|   3669094.7540 |         2009 |             1 |
|   4238002.2831 |         2008 |             1 |
|   3760427.0833 |         2007 |             1 |
|   4548565.1515 |         2006 |             1 |
|   4450312.5000 |         2005 |             1 |
|   6003369.5652 |         2004 |             1 |
|   5235000.0000 |         2003 |             1 |
|   4578787.8788 |         2002 |             1 |
|   4831707.3171 |         2001 |             1 |
|   3743396.2264 |         2000 |             1 |
|   6550000.0000 |         1999 |             1 |
|   5582608.6957 |         1997 |             1 |
|   5212581.0000 |         1995 |             1 |
+----------------+--------------+---------------+
```

Q3. What are each position as well as handedness' expected salaries?

The results reveal that right-handed Left Wing players boast the highest overall average salaries, followed by left-handed Right Wing players in the second position. Unexpectedly, Goaltenders earn the least overall, with right-handed Goaltenders earning even less than their left-handed counterparts. Generally speaking, right-handed players appear to have a comparative advantage in terms of compensation, as they make up slightly more than half of the proportion in the top five categories.

```
mysql> select avg(s.salary) as average_salary, p.position, p.handedness from salary s join player p on s.player_id = p.player_id where position
 is not null and handedness is not null group by p.position, p.handedness order by average_salary desc;
+----------------+----------+------------+
| average_salary | position | handedness |
+----------------+----------+------------+
|   3007826.0516 | LW       | R          |
|   2692803.7143 | RW       | L          |
|   2289886.6395 | C        | L          |
|   2284128.3771 | D        | R          |
|   2060940.1087 | C        | R          |
|   2045984.3969 | G        | L          |
|   2034368.5238 | D        | L          |
|   2031911.4299 | RW       | R          |
|   1961625.1140 | LW       | L          |
|   1398271.2766 | G        | R          |
+----------------+----------+------------+
10 rows in set (0.11 sec)
```

Q4. What are the numbers of counts of players of each nationality?

I start thinking about this question by checking the unique nationalities included in our dataset and there are 25 of them.

```
mysql> select count(distinct nationality) as nationality_count from player;
+--------------------+
| nationality_count |
+--------------------+
|                 25 |
+--------------------+
1 row in set (0.04 sec)
```

Next, I analyze the distribution of players based on their nationality. It's evident that there is a significantly higher number of Canadian players compared to other nationalities, with the United States remaining in the top two positions, as anticipated. It's important to note that, when interpreting this result, we should be mindful that, during our data cleaning process, numerous European players were excluded because our server lacked the capability to match their names and generate corresponding player IDs for data analysis! Thus, we should not take the result as globally applicable until we have a ratherly complete dataset with European players being properly represented.

```
mysql> select count(1) as count_of_players, nationality from player where nationality is not null group by nationality order by count(1) desc;
+------------------+-------------+
| count_of_players | nationality |
+------------------+-------------+
|             1359 | CAN         |
|              769 | USA         |
|              253 | SWE         |
|              151 | FIN         |
|              149 | RUS         |
|              124 | CZE         |
|               38 | SVK         |
|               33 | CHE         |
|               27 | DEU         |
|               17 | DNK         |
|               15 | LVA         |
|               11 | BLR         |
|                7 | FRA         |
|                7 | AUT         |
|                3 | UKR         |
|                3 | NOR         |
|                3 | GBR         |
|                2 | LTU         |
|                2 | SVN         |
|                1 | KAZ         |
|                1 | BHS         |
|                1 | NGA         |
|                1 | NGA         |
|                1 | AUS         |
|                1 | HRV         |
|                1 | NLD         |
+------------------+-------------+
25 rows in set (0.04 sec)
```

Q5. Do Canadian hockey teams tend to have more Canadian players?

Building upon the preceding question, where we established a notable abundance of Canadian players compared to players of other nationalities, we extend our inquiry. We hypothesize that Canadian teams might exhibit a tendency to have more Canadian players, possibly to facilitate communication in both French and English. Among the 15 teams considered, 6 are Canadian, while the remaining 9 are American. Apparently, Canadian teams are over-representative in

having the most Canadian players, especially considering that there are only 7 Canadian teams among the total of 32 teams in the league.

```
mysql> select count(1) as player_count, p.nationality, p.team_short, t.country from player p join team t on t.team_short = p.team_short where p
.nationality is not null group by p.nationality, p.team_short, t.country order by player_count desc limit 15;
+--------------+-------------+------------+---------+
| player_count | nationality | team_short | country |
+--------------+-------------+------------+---------+
|           70 | CAN         | MTL        | CAN     |
|           65 | CAN         | EDM        | CAN     |
|           62 | CAN         | ANA        | USA     |
|           56 | CAN         | TOR        | CAN     |
|           55 | CAN         | OTT        | CAN     |
|           54 | CAN         | COL        | USA     |
|           53 | CAN         | CHI        | USA     |
|           52 | CAN         | ARI        | USA     |
|           49 | CAN         | PHI        | USA     |
|           48 | CAN         | TBL        | USA     |
|           48 | CAN         | MIN        | USA     |
|           47 | CAN         | VAN        | CAN     |
|           45 | CAN         | NYI        | USA     |
|           44 | CAN         | CGY        | CAN     |
|           41 | CAN         | PIT        | USA     |
+--------------+-------------+------------+---------+
15 rows in set (0.05 sec)
```

Q6. What is the maximum second stayed on ice grouped by player, team, year of game, and division?

By grouping data based on players, teams, game years, and divisions, we identify that the maximum time spent on ice is 1646 seconds, equivalent to 27 minutes and 26 seconds in a single game. Notably, three players – Drew Doughty, Ryan Suter, and Erik Karlsson – appear multiple times on the list of players with the highest time spent on ice during a game. This repetition supports the presence of their respective teams and affiliated divisions on the list multiple times. Curiously, the absence of any player from the Atlantic Division in the top time spent on ice category sparks intrigue. One potential explanation could be attributed to the superior collaboration among players within the Atlantic teams!

```
mysql> select max(ps.average_min_on_ice*60+ps.average_sec_on_ice) as max_second_on_ice, ps.year, ps.player_id, p.name, p.team_short, t.division
, d.name, d.conference from player_stat ps join player p on ps.player_id = p.player_id join team t on t.team_short = p.team_short join division
 d on t.division = d.division_id group by p.player_id, ps.year, p.name, p.team_short, t.division, d.name, d.conference having max_second_on_ice
 > 0 order by max_second_on_ice desc limit 10;
+------------------+------+-----------+-----------------+------------+----------+-------------+------------+
| max_second_on_ice | year | player_id | name            | team_short | division | name        | conference |
+------------------+------+-----------+-----------------+------------+----------+-------------+------------+
|             1646 | 2017 | 8470834   | Dustin Byfuglien | WPG        |       25 | Central     | Western    |
|             1628 | 2017 | 8474563   | Drew Doughty    | LAK        |       26 | Pacific     | Western    |
|             1615 | 2017 | 8470600   | Ryan Suter      | DAL        |       25 | Central     | Western    |
|             1610 | 2018 | 8474563   | Drew Doughty    | LAK        |       26 | Pacific     | Western    |
|             1610 | 2017 | 8474578   | Erik Karlsson   | PIT        |       24 | Metropolitan | Eastern    |
|             1606 | 2018 | 8470600   | Ryan Suter      | DAL        |       25 | Central     | Western    |
|             1604 | 2018 | 8474578   | Erik Karlsson   | PIT        |       24 | Metropolitan | Eastern    |
|             1601 | 2019 | 8470600   | Ryan Suter      | DAL        |       25 | Central     | Western    |
|             1595 | 2019 | 8474563   | Drew Doughty    | LAK        |       26 | Pacific     | Western    |
|             1590 | 2011 | 8474590   | John Carlson    | WSH        |       24 | Metropolitan | Eastern    |
+------------------+------+-----------+-----------------+------------+----------+-------------+------------+
10 rows in set (0.28 sec)
```

Q7. How many times of championships do teams that have won the Stanley Cup win?

Based on the displayed results, a total of 17 teams have secured at least one Stanley Cup since 1990, suggesting a relatively concentrated distribution among the 32 teams over the 33 years.

Notably, six of these 17 teams have clinched the championship more than three times, underscoring their high level of competitiveness regardless of player composition changes across years.

```
mysql> select sc.winning_team, count(1) as time_winning_champ from stanley_cup sc where sc.winning_team is not null group by sc.winning_team;
+---------------------+--------------------+
| winning_team        | time_winning_champ |
+---------------------+--------------------+
| Anaheim Ducks       |                  1 |
| Boston Bruins       |                  1 |
| Carolina Hurricanes |                  1 |
| Chicago Blackhawks  |                  3 |
| Colorado Avalanche  |                  3 |
| Dallas Stars        |                  1 |
| Detroit Red Wings   |                  4 |
| Edmonton Oilers     |                  1 |
| Los Angeles Kings   |                  2 |
| Montreal Canadiens  |                  1 |
| New Jersey Devils   |                  3 |
| New York Rangers    |                  1 |
| Pittsburgh Penguins |                  5 |
| St. Louis Blues     |                  1 |
| Tampa Bay Lightning |                  3 |
| Vegas Golden Knights|                  1 |
| Washington Capitals |                  1 |
+---------------------+--------------------+
17 rows in set (0.05 sec)
```

Q8. What is the saving percentage of the team in the year they win the championship?

As indicated by the results below, five out of the seven Stanley Cup-winning teams boast average saving percentages exceeding 0.9. In contrast, the Tampa Bay Lightning and the Colorado Avalanche had saving percentages close to 0.9 in the years 2021 and 2022. This observation suggests that in the 2021 and 2022 seasons, all teams demonstrated a higher level of competitiveness.

```
mysql> select sc.winning_team, sc.year, avg(ps.save_percentage) from stanley_cup sc join team t on t.team_full = sc.winning_team join player p
on p.team_short = t.team_short join player_stat ps on p.player_id = ps.player_id where sc.winning_team is not null and sc.game_id is not null a
nd ps.save_percentage is not null and ps.games_played > 0 and ps.year = sc.year group by sc.winning_team, sc.year;
+---------------------+------+------------------------+
| winning_team        | year | avg(ps.save_percentage) |
+---------------------+------+------------------------+
| Pittsburgh Penguins | 2017 |      0.9434000015258789 |
| Washington Capitals | 2018 |       0.914000004529953 |
| St. Louis Blues     | 2019 |      0.9269999861717224 |
| Tampa Bay Lightning | 2020 |      0.9042499959468842 |
| Tampa Bay Lightning | 2021 |      0.8922000050544738 |
| Colorado Avalanche  | 2022 |       0.897599995136261 |
| Vegas Golden Knights| 2023 |      0.9193333387374878 |
+---------------------+------+------------------------+
7 rows in set (0.06 sec)
```

Section III.

Data from diverse online sources are extracted to construct the nine tables. I will systematically introduce each table and its construction process in alphabetical order. The relevant R codes utilized for data retrieval and cleaning, as well as the SQL script for database-building, will be appended for your reference.

1. Table “arena”

The contents of this table encompass details regarding various ice arenas in North America. The table is constructed by extracting data from JSON files obtained through APIs used for the team table. However, the existing table exclusively includes details about the primary ice arenas for the 32 teams, specifically their home arenas. Numerous ice arenas associated with other ice hockey leagues, serving different purposes, and situated in countries other than Canada and the U.S. are not included. I am aware of online resources beyond the scope of this project that contain additional information. If you have an interest in expanding the current arena table, I would be delighted to assist!

2. Table “coach”

The construction of the "coach" table involves leveraging several online resources. Initially, I utilize R for web scraping to extract the list of current NHL head coaches for each team from this Wikipedia page. Subsequently, I refer to both this Wikipedia page and the NHL mobile application to obtain acronyms used in the "served_team" field, specifically the acronyms for the teams. Historical coaches for teams are then web scraped from this website, and a data cleaning process is implemented using R.

Addressing the issue of inconsecutive service in a NHL team for the same coach, including values in “served_team,” "from_date," and "to_date," I individually cross-reference them if not already attached in the raw data. I also look values for “professional_career” up one by one and insert them to the table. Information regarding the "jack_adams_award" is sourced from this webpage. Finally, I export the data frames to CSV files for further data manipulation and subsequent importation into DataGrip.

Some notable calculation formulas include:

a. Points percentage is calculated as follows: (Wins + ½ Ties + ½ Overtime/shootout losses) ÷ Games Coached (not applicable in this database anymore); and
b. length of seasons coaching a team = end_season - start_season + 1.

3. Table “division”

This table holds data regarding divisions and their associated conferences within the NHL. As of 2023, there are 4 divisions, each consisting of 8 teams, spread across two conferences. However, when looking back to the year 1990, numerous alterations have occurred in terms of division names, numbers, and team compositions. To capture the historical evolution, I utilized R programming to scrape HTML tables from a Wikipedia page detailing organizational changes in the NHL. Subsequently, I cleaned the data and exported the resulting data frame to a .csv file for importation into DataGrip.

4. Table “game”

The "game" table stands as a central component in this database, providing comprehensive information on both regular and post-season games. Pre-season games are not included because we both agreed that they’re not of interest to us. Employing APIs and R programming, I retrieve JSON files spanning from the 2013 season to the 2023 season. The decision to start from 2013 is due to limited and less usable information in the APIs for seasons predating 2013, deviating from my initial plan to capture games starting from the 1990 season.

Upon data retrieval, I exclude rows (representing games) that lack scores for either the home or away team. Games reported as "scheduled," "postponed," or "canceled" rather than "closed" are omitted from subsequent MySQL data analysis queries. Additionally, the three "All-Star" games within each regular game season are removed due to their inclusion complicating analysis with a mix of teams.

Subsequent to these exclusions, I conduct data cleaning to ensure alignment with the Entity-Relationship Diagram (ERD) blueprint and export the refined dataset to a .csv file.

5. Table “player”

In contrast to the coach table, the player table may be the simplest in terms of its data collection and manipulation process, yet it holds a central position among all nine tables. This Kaggle dataset furnishes detailed information about 3,070 NHL players who have been active at any point between 2008 and the present. The dataset is consistently updated by the author using data from moneypuck.com, alleviating the need for me to scrape the website myself.

This table captures demographic information for NHL players who meet the aforementioned criteria. Among all the data sources housing player information, this one stands out as the most valuable since it incorporates unified player IDs that I can leverage to establish connections with other tables. Despite some null values in columns such as "state," "year_drafted," "round_drafted," and "draft_overall_rank," which may be irrelevant for some players, the reasons for certain players lacking valid "DOB" and "nationality" are not explicitly disclosed. In this project phase, I opt not to conduct further research on each player to fill in the gaps. However, I remain open to exploring and researching these players in future endeavors.

6. Table “player_stat”

The "Player_stat" table stands out with the highest number of fields and rows among all nine tables in this database, owing to its dynamic nature and time-sensitive content. To construct this table, I scrape data from a website, gathering information from the 2010 season to the 2023 season, and subsequently clean the data using R. I then utilize the "vlookup" function in Excel to match player IDs with the corresponding players.

An inconsistency arises in the representation of some players' names, particularly those of European players with special characters, accents, and name abbreviations, leading to varying displays in the two datasets. For example, Zachary might be denoted as Zach in one dataset, while Joshua may represent the same player recorded as Josh in the other dataset. This naming discrepancy is also evident in the "player_stat" table. Due to time constraints, I have not addressed all instances, as it requires a substantial amount of time that I currently do not have available. To ensure field referencability, I eliminate rows where the algorithm cannot identify exact name matches, subsequently obtaining their corresponding player IDs.

This effort currently excludes a significant number of European players in both the salary and player_stat tables. However, I am enthusiastic about dedicating time to reintegrate them in the future. Following data cleaning, the table comprises 15,891 rows.

7. Table “salary”

The salary table stands as the sole table featuring solely numeric values without any accompanying characters. Data is scraped from a website using R, encompassing information

related to professional performance, demographic details, career timelines, and more based on users' selections, spanning from the 2010 season to the 2023 season. Subsequently, I performed data cleaning and mutated corresponding player IDs for salary rows using the "vlookup" function in Excel and cross-referencing with the player table.

An inconsistency arises in the display of some players' names, particularly those of European players with special characters or accents and name abbreviations, resulting in different representations in the two datasets. For instance, Zachary might be denoted as Zach in one dataset, while Joshua may represent the same player recorded as Josh in the other dataset. This naming discrepancy is also observed in the player_stat table. Due to time constraints, I have not addressed all instances, as it requires a significant amount of time that I currently do not have available. To ensure referencability of the fields, I eliminate rows where the algorithm cannot identify exact name matches and subsequently obtain their corresponding player IDs. This effort currently omits a significant number of European players in both the salary and player_stat tables. I am eager to allocate time to reintegrate them in the future.

Following data cleaning, there are a total of 15,291 rows of stored data.

8. Table “stanley_cup”

The Stanley Cup is the prestigious trophy presented annually to the playoff champion of the National Hockey League (NHL). In the Stanley Cup Finals, the team that secures victory in four games first is crowned the Stanley Cup champion for that particular season. All post-season games, commonly referred to as playoffs, contribute to the history of the Stanley Cup. However, for the sake of dataset readability, the "stanley_cup" table exclusively incorporates information from the last series of games between the championship teams and their opponents.

To construct this table, I employed data scraping from a website to extract details such as championships, opponents, counts of games won in the series, and corresponding years using R. After completing data cleaning to ensure proper interpretation of special symbols in R, I then retrieved trustee information from a Wikipedia page, finalizing the table-building process.

While attempting to mutate the game IDs for each Stanley Cup game in the table, I encountered challenges due to incomplete information about championship playoffs within the game table. Consequently, I had no alternative but to retain some games with blank IDs. In addition, the 2004-2005 NHL lockout results in the null values in the “winning_team” and the “losing_team” of year 2005.

9. Table “team”

The team table serves as the final pivotal and foundational element of this dataset. Given the overarching role of teams in the league, most tables are expected to establish at least one connection with this table. I employ APIs to fetch JSON files for NHL teams and then perform data cleaning in R, preparing the data for importation into DataGrip as a CSV file. Initially, I included historical data for each team, encompassing changes in city locations, franchise names, division affiliations, and arenas. However, complications arose when connecting foreign keys, prompting me to retain a single row for each team based on their current information.

Other online resources used during the secondary research process for table building are listed below.

1. Introduction to Ice Hockey Statistics
2. History of NHL Conferences and Divisions

Section IV.
The ERD is attached to the last page of this document.

I appreciate every patience you have given to us throughout this semester. As always, please let me know if you have any questions. I hope to keep in touch with you!

Let's go Leafs,
Chia Wen

division in NHL

primary key	division_id	int unsigned	unique id for each division	can't be NULL
	from_year	year	the starting year of the conference and the division	can't be NULL
	name	varchar(255)	name of the division	can't be NULL
	conference	varchar(255)	name of the conference the division is affiliated with	can't be NULL

player in NHL

primary key	record_id	int unsigned	unique id for each row of data	can't be NULL
foreign key	player_id	varchar(255)	unique 7-digit id for each player	can't be NULL
	name	varchar(255)	full name of the player	can't be NULL
	DOB	date	date of birth of the player	can be NULL
	city	varchar(255)	city the player was born	can be NULL
	state	char(2)	state the player was born	can be NULL
	nationality	char(3)	nationality of the player (if dual, list only one due to the data source)	can be NULL
	weight	int	weight in pounds of the player	can be NULL
	height_inch	int	height in inches of the player	can be NULL
	handedness	varchar(5)	player's handedness	can be NULL
	year_drafted	year	the year the player was drafted by an NHL team	can be NULL
	round_drafted	int	the round the player was drafted by an NHL team	can be NULL
	draft_overall_rank	int	the overall ranking of the player in their draft year	can be NULL
foreign key	team_short	varchar(3)	3-digit abbreviation of the team the player is in	can be NULL
	position	varchar(10)	the position the player plays	can be NULL
	number	int	the number of the player in the team	can be NULL

coach in NHL

primary key	id	int unsigned	unique id for each row of data	can't be NULL
	coach_id	int unsigned	unique id for each coach	can't be NULL
	first_name	varchar(255)	first name of the coach	can't be NULL
	last_name	varchar(255)	last name of the coach	can't be NULL
	active_in_2023	bit	active in 2023 = true and false otherwise	can't be NULL
	start_season	year	the starting year of the season the coach started serving the team	can't be NULL
	end_season	year	the starting year of the season the coach ended serving the team	can't be NULL
foreign key	served_team	varchar(3)	3-digit abbreviation of the team the coach serve(s/d)	can't be NULL
	from_date	date	the date the coach started serving the team	can't be NULL
	to_date	date	the date the coach ended serving the team	can't be NULL
	jack_adams_award	bit	won Jack Adams Award during the period of service = true and false other wse	can't be NULL
	professional_career	varchar(255)	the professional playing career of the coach	can be NULL

salary in NHL

primary key	record_id	int unsigned	unique id for each row of the data	can't be NULL
foreign key	player_id	varchar(255)	7-digit unique id composed of numbers for identifying each player	can be NULL (due to differences of informaiton stored on the same player in multiple data sources)
	salary	int	number of salary for the player	can't be NULL
	from_season	year	the starting year of the salary level (should be from 2010 to 2023 due to the dataset I found)	can't be NULL
	to_season	year	the starting year of the salary level (should be from 2010 to 2023 due to the dataset I found)	can't be NULL

team in NHL

primary key	id	int unsigned	unique id for each row of data	can't be NULL
	team_id	int unsigned	unique id for each team	can't be NULL
	from_year	year	the year the team started like this	can't be NULL
	to_year	year	the year the team ended like this	can't be NULL
	nickname	varchar(255)	nickname of the team	can't be NULL
foreign key	team_full	varchar(255)	full name including city and nickname of the team	can't be NULL
foreign key	team_short	varchar(3)	3-digit acronym of the team	can't be NULL
	city	varchar(255)	city name the team plays home games	can't be NULL
	state	char(2)	2-digit state acronym the team plays home games	can't be NULL
	country	char(3)	3-digit country acronym the team plays home games	can't be NULL
	timezone	varchar(255)	timezone the team is located in	can't be NULL
foreign key	division	int	unique id for each division	can't be NULL
foreign key	home_arena	int	id of the home arena the team plays in	can be NULL

arena in NHL

primary key	arena_id	int unsigned	unique id for each arena	can't be NULL
	arena_name	varchar(255)	name of the arena in texts	can't be NULL
	capacity	int	number of capacity of the arena	can't be NULL
	address	varchar(255)	address including the numbers and the street names of the arena location	can't be NULL
	state	char(2)	2-digit state abbreviation of the arena location	can be NULL
	zip	varchar(10)	zip code of the arena location	can be NULL
	country	char(3)	3-digit country abbreviation in where the arena is located in	can't be NULL
	timezone	varchar(255)	timezone the arena uses	can't be NULL

player_stat in NHL

primary key	record_id	int unsigned	unique id for each row of data	can't be NULL
foreign key	payer_id	varchar(255)	unique 7-digit id for each player	can be NULL
	player_name	varchar(255)	full name of the player (leave this field instead of relying fully on player_id because different data sources may list names differently and thus player_id may be NULL after data manipulation)	can't be NULL
	games_played	int	number of games the player played in that season	can be NULL
	goals	int	number of goals the player scored in that season	can be NULL
	assists	int	number of assists the player scored in that season	can be NULL
	points	int	number of points the player scored in that season	can be NULL
	plus/minus	int	the number of team even strength or shorthanded goals for minus the number of team even strength or shorthanded goals against while the player is on the ice	can be NULL
	shots_on_goal	int	number of shots on goal the player scored in that season	can be NULL
	shooting_percentage	float	shooting percentage of the player in that season	can be NULL
	points_per_game	float	points per game the player scored in that season	can be NULL
	average_min_on_ice	int	player's average time of minute on ice in that season	can be NULL
	average_sec_on_ice	int	player's average time of second on ice in that season	can be NULL
	individual_expected_goals	float	player's expected goals in that season	can be NULL
	individual_shots_on_goal	int	player's shots on goal in that season	can be NULL
	individual_corsi	int	the sum of shot attempts on net made by this player in that season including missed and blocked shots	can be NULL
	individual_fenwick	int	the sum of shot attempts on net made by this player in that season including missed shots	can be NULL
	individual_expected_goals_per60min	float	an estimate of the total goals a player is expected to score per 60 minutes of ice time (dividing this value by the plaers average time on ice per game is an estimate of the players expected goals per game)	can be NULL
	individual_shots_on_goal_per60min	float	individual shots on goal per 60 minutes of ice time	can be NULL
	individual_corsi_per60min	float	the number of shot attempts on net made by this player including missed shots and blocked shots per 60 minutes of ice time	can be NULL
	individual_fenwick_per60min	float	the number of shot attempts on net made by this player including missed shots (excluding blocked shots) per 60 minutes of ice time	can be NULL
	wins	int	games the goaltender has won the current season	can be NULL
	loses	int	games the goaltender has lost (A goaltender is credited with a win or loss when he is either on the ice when – or was pulled for an extra attacker immediately before – the game-winning goal was scored)	can be NULL
	shutouts	int	number of games where the goaltender had no goals against him and was the only goaltender from his team to play in the game	can be NULL
	goals_against_average	float	mean goals-per-60 minutes scored on the goaltender	can be NULL
	saving_percentage	float	percentage of the total shots faced the goaltender has saved	can be NULL
	goals_against_per60min	float	goals against per 60 minutes of ice time	can be NULL
	goaltender_related_expected_goals_against_per60min_of_ice_time	float	an estimate of the amount of goals against per 60 minutes of ice time expected based on various attributes of all fenick shots taken against that goalie (distance, angle, type, score state, etc.)	can be NULL
	goals_saved_above_expected_per60min_of_ice_time	float	expected goals against per 60 minus goals against per 60	can be NULL
	year	year	the ending year of the game season of the data scraped	can't be NULL

game in NHL

primary key	record_id	int unsigned	unique id for each row of data	can't be NULL
foreign key	game_id	int	unique id for each game	can't be NULL
	date	date	date the game was played	can be NULL
foreign key	home_short	varchar(3)	3-digit acronym of the home team	can be NULL
foreign key	away_short	varchar(3)	3-digit acronym of the away team	can be NULL
foreign key	winner	varchar(3)	3-digit acronym of the winner team	can be NULL
	winner_score	int	the score of the winner team	can be NULL
foreign key	rivalry	varchar(3)	3-digit acronym of the losing team	can be NULL
	rivalry_score	int	the score of the losing team	can be NULL
	playoff	bit	playoff game (i.e., post-season game) = true and regular game = false (no pre-season games included)	can be NULL

stanley_cup in NHL

primary key	id	int unsigned	unique id for each row of data	can't be NULL
foreign key	game_id	int	unique id for each game (the same as in the game table)	can be NULL
	year	year	year of the stanley cup game	can't be NULL
foreign key	winning_team	varchar(255)	full name of the winning team	can be NULL
foreign key	losing_team	varchar(255)	full name of the losing team	can be NULL
	champion_wins_loses	varchar(255)	numbers of wins and loses of the winning team in this playoff series	can be NULL
	trustee_1	varchar(255)	full name of the first trustee	can't be NULL
	trustee_2	varchar(255)	full name of the second trustee	can't be NULL

Appendix

A. DDL for table-building from DataGrip

```
create table if not exists arena
(
    arena_id   int auto_increment
        primary key,
    arena_name varchar(255) not null,
    capacity   int          not null,
    address    varchar(255) not null,
    state      char(2)      null,
    zip        varchar(10)  null,
    country    char(3)      not null,
    timezone   varchar(255) not null,
    constraint arena_name_index
        unique (arena_name)
);

create table if not exists division
(
    division_id int          not null
        primary key,
    from_year   year         not null,
    name        varchar(255) not null,
    conference  varchar(255) not null
);

create index division_name_index
    on division (name);

create index division_year_index
    on division (from_year);

create table if not exists team
(
    id         int auto_increment
        primary key,
    team_id    int          not null,
    from_year  year         not null,
    to_year    year         not null,
    nickname   varchar(255) not null,
    team_full  varchar(255) not null,
    team_short varchar(3)   not null,
    city       varchar(255) not null,
    state      char(2)      not null,
    country    char(3)      not null,
```

```
    timezone   varchar(255) not null,
    division   int          not null,
    home_arena int          null,
    constraint team_full_index
        unique (team_full),
    constraint team_short_index
        unique (team_short),
    constraint team_arena__fk
        foreign key (home_arena) references arena (arena_id),
    constraint team_division__fk
        foreign key (division) references division
(division_id)
);

create table if not exists coach
(
    id                  int auto_increment
        primary key,
    coach_id            int          not null,
    first_name          varchar(255) not null,
    last_name           varchar(255) not null,
    active_in_2023      bit          not null,
    start_season        year         not null,
    end_season          year         not null,
    served_team         varchar(3)   not null,
    from_date           date         not null,
    to_date             date         not null,
    jack_adams_award    bit          not null,
    professional_career varchar(255) null,
    constraint coach_team__fk
        foreign key (served_team) references team (team_short)
);

create table if not exists game
(
    record_id     int auto_increment
        primary key,
    game_id       int        not null,
    date          date       not null,
    home_short    varchar(3) not null,
    away_short    varchar(3) not null,
    winner        varchar(3) not null,
    winner_score  int        not null,
    rivalry       varchar(3) not null,
    rivalry_score int        not null,
```

```
    playoff       bit        not null,
    constraint game_team1__fk
        foreign key (winner) references team (team_short),
    constraint game_team2__fk
        foreign key (rivalry) references team (team_short),
    constraint game_team3__fk
        foreign key (home_short) references team (team_short),
    constraint game_team4__fk
        foreign key (away_short) references team (team_short)
);

create index game_id_index
    on game (game_id);

create table if not exists player
(
    record_id          int auto_increment
        primary key,
    player_id          varchar(255) not null,
    name               varchar(255) not null,
    DOB                date         null,
    city               varchar(255) null,
    state              char(2)      null,
    nationality        char(3)      null,
    weight             int          null,
    height_inch        int          null,
    handedness         varchar(5)   null,
    year_drafted       year         null,
    round_drafted      int          null,
    draft_overall_rank int          null,
    team_short         char(3)      null,
    position           varchar(10)  null,
    number             int          null,
    constraint player_id_index
        unique (player_id),
    constraint player_team__fk
        foreign key (team_short) references team (team_short)
);

create index player_team_index
    on player (team_short);

create index player_year_index
    on player (year_drafted);
```

```
create table if not exists player_stat
(
    record_id
int          not null
        primary key,
    player_id
varchar(255) not null,
    player_name
varchar(255) not null,
    games_played
int          null,
    goals
int          null,
    assists
int          null,
    points
int          null,
    points_per_game
float        null,
    `plus/minus`
int          null,
    shots_on_goal
int          null,
    shooting_percentage
float        null,
    average_min_on_ice
int          null,
    average_sec_on_ice
int          null,
    individual_expected_goals
float        null,
    individual_shots_on_goal
int          null,
    individual_corsi
int          null,
    individual_fenwick
int          null,
    individual_expected_goals_per60min
float        null,
    individual_shots_on_goal_per60min
float        null,
    individual_corsi_per60min
float        null,
    individual_fenwick_per60min
float        null,
```

```
    wins
int          null,
    loses
int          null,
    shutouts
int          null,
    goals_against_average
float        null,
    save_percentage
float        null,
    goals_against_per60min_of_ice_time
float        null,

goaltender_related_expected_goals_against_per60min_of_ice_time
float        null,
    goals_saved_above_expected_per60min_of_ice_time
float        null,
    year
year         not null,
    constraint player_stat_player__fk
        foreign key (player_id) references player (player_id)
);

create index player_stat_player_index
    on player_stat (player_id, player_name);

create table if not exists salary
(
    record_id   int          not null
        primary key,
    player_id   varchar(255) not null,
    salary      int          not null,
    from_season year         not null,
    to_season   year         not null,
    constraint salary_player__fk
        foreign key (player_id) references player (player_id)
);

create index salary_player_index
    on salary (player_id);

create table if not exists stanley_cup
(
    id                  int auto_increment
        primary key,
```

```
    game_id             int          null,
    year                year         not null,
    winning_team        varchar(255) null,
    losing_team         varchar(255) null,
    champion_wins_loses varchar(255) null,
    trustee_1           varchar(255) not null,
    trustee_2           varchar(255) not null,
    constraint stanley_cup_game_game_id_fk
        foreign key (game_id) references game (game_id),
    constraint stanley_cup_lose__fk
        foreign key (losing_team) references team (team_full),
    constraint stanley_cup_teams__fk
        foreign key (winning_team) references team (team_full)
);

create index stanley_cup_game_index
    on stanley_cup (game_id);

create index stanley_cup_year_index
    on stanley_cup (year);

create index team_division_index
    on team (division);

create index team_year_index
    on team (from_year, to_year);
```

B. Web-scraping and data-cleaning codes from R

```
library(xml2)
library(rvest)
library(tidyverse)
library(dplyr)
library(plyr)
library(stringr)
library(tidyr)
library(RCurl)
library(purrr)
library(dbplyr)
library(jsonlite)
# for division table
conference <-
read_html("https://en.wikipedia.org/wiki/History_of_organizational_ch
anges_in_the_NHL")
str(conference)
# select the table using CSS selector
```

```
nodes_conference <- html_nodes(conference, "table")
# extract the table content using subsetting
con_90 <- html_table(nodes_conference)[[41]]
con_91 <- html_table(nodes_conference)[[43]]
con_92 <- html_table(nodes_conference)[[45]]
con_93 <- html_table(nodes_conference)[[47]]
con_95 <- html_table(nodes_conference)[[48]]
con_96 <- html_table(nodes_conference)[[49]]
con_97 <- html_table(nodes_conference)[[50]]
con_98 <- html_table(nodes_conference)[[52]]
con_99 <- html_table(nodes_conference)[[54]]
con_00 <- html_table(nodes_conference)[[56]]
con_06 <- html_table(nodes_conference)[[57]]
con_11 <- html_table(nodes_conference)[[58]]
con_13 <- html_table(nodes_conference)[[59]]
con_14 <- html_table(nodes_conference)[[60]]
con_17 <- html_table(nodes_conference)[[62]]
con_20 <- html_table(nodes_conference)[[63]]
con_21 <- html_table(nodes_conference)[[65]]
write.csv(con_90, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1990.csv", row.names = FALSE)
write.csv(con_91, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1991.csv", row.names = FALSE)
write.csv(con_92, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1992.csv", row.names = FALSE)
write.csv(con_93, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1993.csv", row.names = FALSE)
write.csv(con_95, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1995.csv", row.names = FALSE)
write.csv(con_96, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1996.csv", row.names = FALSE)
write.csv(con_97, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1997.csv", row.names = FALSE)
write.csv(con_98, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1998.csv", row.names = FALSE)
```

```
write.csv(con_99, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_1999.csv", row.names = FALSE)
write.csv(con_00, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2000.csv", row.names = FALSE)
write.csv(con_06, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2006.csv", row.names = FALSE)
write.csv(con_11, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2011.csv", row.names = FALSE)
write.csv(con_13, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2013.csv", row.names = FALSE)
write.csv(con_14, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2014.csv", row.names = FALSE)
write.csv(con_17, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2017.csv", row.names = FALSE)
write.csv(con_20, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2020.csv", row.names = FALSE)
write.csv(con_21, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/conference_2021.csv", row.names = FALSE)

# for coach table
coach <-
read_html("https://en.wikipedia.org/wiki/List_of_NHL_head_coaches")
nodes_coach <- html_nodes(coach, "table")
headcoach <- html_table(nodes_coach)[[2]]
headcoach1 <- headcoach[-1, -c(10, 17)]
colnames(headcoach1) <- c("served_team", "coach_name", "from_date",
"team_games_coached", "team_wins", "team_losses", "team_ties",
"team_OT/SO_losses", "team_points", "career_games_coached",
"career_wins", "career_losses", "career_ties", "career_OT/SO_losses",
"career_points", "professional_career")
write.csv(headcoach1, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/coach/headcoach1.csv", row.names = FALSE)
## for historical coach data from an interactive website
```

```
registered_coach <-
read_html("https://records.nhl.com/registry/head-coach-regular-season
")

### parse extracted string
registered_coach_json <- fromJSON("G:/My Drive/0. study
abroad/academic/10. 2023 Fall/3. SI 564 SQL &
Databases/Homework/Final Project/0.
Datasets/tables/coach/coach-franchise-records.json")
### change column names to avoid the leading "data." and "$" for the
coach_id column
registered_coach_df <- as.data.frame(registered_coach_json)
colnames(registered_coach_df) <-
sub("data.","",colnames(registered_coach_df))
colnames(registered_coach_df[, 3]) <- "coach_id"
### edit values in columns "endSeason" and "startSeason"
registered_coach_df$endSeason <-
paste(substr(as.character(registered_coach_df$endSeason), 1, 4), "-",
substr(as.character(registered_coach_df$endSeason), 5, 8), sep = "")
registered_coach_df$startSeason <-
paste(substr(as.character(registered_coach_df$startSeason), 1, 4),
"-", substr(as.character(registered_coach_df$startSeason), 5, 8), sep
= "")
### drop rows that have not been active since the season of 1990
registered_coach_df_1 <- subset(registered_coach_df,
as.numeric(substr(registered_coach_df$endSeason, 6, 9)) <= 1990)
registered_coach_df_2 <-
registered_coach_df[setdiff(rownames(registered_coach_df),
rownames(registered_coach_df_1)), ]
### export the file to csv
write.csv(registered_coach_df_2, "G:/My Drive/0. study
abroad/academic/10. 2023 Fall/3. SI 564 SQL &
Databases/Homework/Final Project/0.
Datasets/tables/coach/registered_coach.csv")


# for game table
##2022-23 regular
game_reg22 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg22.json")
game_reg22_venue <- as.data.frame(game_reg22$games$venue)
game_reg22_venue[, 11] <- c(1:1319)
colnames(game_reg22_venue)[11] <- "id_2"
game_reg22_home <- as.data.frame(game_reg22$games$home)
```

```
game_reg22_home[, 6] <- c(1:1319)
colnames(game_reg22_home)[6] <- "id_2"
game_reg22_away <- as.data.frame(game_reg22$games$away)
game_reg22_away[, 6] <- c(1:1319)
colnames(game_reg22_away)[6] <- "id_2"
game_reg22_game <- as.data.frame(game_reg22$games)
game_reg22_game <- game_reg22_game[, -c(9:12)]
game_reg22_game[, 10] <- c(1:1319)
colnames(game_reg22_game)[10] <- "id_2"
game_reg22_all <- merge(game_reg22_game, game_reg22_venue, by =
"id_2")
game_reg22_all <- merge(game_reg22_all, game_reg22_home, by = "id_2")
game_reg22_all <- merge(game_reg22_all, game_reg22_away, by = "id_2")
game_reg22_all <- game_reg22_all%>% relocate(title, .before = status)
game_reg22_all <- game_reg22_all%>% relocate(state, .before =
country)
game_reg22_all <- game_reg22_all%>% relocate(zip, .before = country)
game_reg22_all$playoff <- 0
colnames(game_reg22_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game",  "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"away_id", "away_name", "away_short", "sr_id_away", "reference_away",
"playoff")


##2022-23 postseason
game_pst22 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post22.json")
game_pst22_venue <- as.data.frame(game_pst22$games$venue)
game_pst22_venue[, 11] <- c(1:105)
colnames(game_pst22_venue)[11] <- "id_2"
game_pst22_home <- as.data.frame(game_pst22$games$home)
game_pst22_home[, 7] <- c(1:105)
colnames(game_pst22_home)[7] <- "id_2"
game_pst22_away <- as.data.frame(game_pst22$games$away)
game_pst22_away[, 7] <- c(1:105)
colnames(game_pst22_away)[7] <- "id_2"
game_pst22_game <- as.data.frame(game_pst22$games)
game_pst22_game <- game_pst22_game[, -c(10:13)]
game_pst22_game[, 10] <- c(1:105)
colnames(game_pst22_game)[10] <- "id_2"
```

```
game_pst22_all <- merge(game_pst22_game, game_pst22_venue, by =
"id_2")
game_pst22_all <- merge(game_pst22_all, game_pst22_home, by = "id_2")
game_pst22_all <- merge(game_pst22_all, game_pst22_away, by = "id_2")
game_pst22_all$playoff <- 1
colnames(game_pst22_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"home_seed", "away_id", "away_name", "away_short", "sr_id_away",
"reference_away", "away_seed", "playoff")

##combine 2022-23 regular and postseason
game22 <- rbind.fill(game_reg22_all, game_pst22_all)
##write to .csv
write.csv(game22, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game22.csv")

##data manipulation
game22_SQL <- game22
game22_SQL$home_short <- gsub("SJ", "SJS", game22_SQL$home_short)
game22_SQL$away_short <- gsub("SJ", "SJS", game22_SQL$away_short)
game22_SQL$home_short <- gsub("TB", "TBL", game22_SQL$home_short)
game22_SQL$away_short <- gsub("TB", "TBL", game22_SQL$away_short)
game22_SQL$home_short <- gsub("LA", "LAK", game22_SQL$home_short)
game22_SQL$away_short <- gsub("LA", "LAK", game22_SQL$away_short)
game22_SQL$home_short <- gsub("NJ", "NJD", game22_SQL$home_short)
game22_SQL$away_short <- gsub("NJ", "NJD", game22_SQL$away_short)
game22_SQL$home_short <- gsub("FLAK", "FLA", game22_SQL$home_short)
game22_SQL$away_short <- gsub("FLAK", "FLA", game22_SQL$away_short)
game22_SQL$winner <- ifelse(game22_SQL$home_points >
game22_SQL$away_points, game22_SQL$home_short,
ifelse(game22_SQL$home_points == game22_SQL$away_points, "tie",
game22_SQL$away_short))
game22_SQL$winner_score <- ifelse(game22_SQL$winner ==
game22_SQL$home_short, game22_SQL$home_points,
ifelse(game22_SQL$winner == "tie", game22_SQL$home_points,
game22_SQL$away_points))
game22_SQL$rivalry <- ifelse(game22_SQL$winner ==
game22_SQL$home_short, game22_SQL$away_short,
ifelse(game22_SQL$winner == "tie", "tie", game22_SQL$home_short))
```

```
game22_SQL$rivalry_score <- ifelse(game22_SQL$winner ==
game22_SQL$home_short, game22_SQL$away_points,
ifelse(game22_SQL$winner == "tie", game22_SQL$home_points,
game22_SQL$home_points))
game22_SQL_1 <- subset(game22_SQL, home_points != 0)
#game22_SQL_1$playoff[grepl("all-star",
tolower(game22_SQL_1$game_title))] <- 0
game22_SQL_1 <- subset(game22_SQL_1, !grepl("all-star",
tolower(game_title)))
game22_SQL_2 <- subset(game22_SQL_1, select = -c(1:5, 7:9, 11:22,
24:27, 29:30, 32:33))
game22_SQL_2 <- game22_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game22_SQL_2 <- game22_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game22_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game22_cleaned.csv")
game22_venue <- game22_SQL_1[, c(11:20)]
write.csv(game22_venue, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game22_venue.csv")

##2021-22 regular
game_reg21 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg21.json")
game_reg21_venue <- as.data.frame(game_reg21$games$venue)
game_reg21_venue[, 11] <- c(1:1417)
colnames(game_reg21_venue)[11] <- "id_2"
game_reg21_home <- as.data.frame(game_reg21$games$home)
game_reg21_home[, 6] <- c(1:1417)
colnames(game_reg21_home)[6] <- "id_2"
game_reg21_away <- as.data.frame(game_reg21$games$away)
game_reg21_away[, 6] <- c(1:1417)
colnames(game_reg21_away)[6] <- "id_2"
game_reg21_game <- as.data.frame(game_reg21$games)
game_reg21_game <- game_reg21_game[, -c(9:12)]
game_reg21_game[, 10] <- c(1:1417)
colnames(game_reg21_game)[10] <- "id_2"
game_reg21_all <- merge(game_reg21_game, game_reg21_venue, by =
"id_2")
game_reg21_all <- merge(game_reg21_all, game_reg21_home, by = "id_2")
game_reg21_all <- merge(game_reg21_all, game_reg21_away, by = "id_2")
game_reg21_all <- game_reg21_all%>% relocate(title, .before = status)
```

```
game_reg21_all$playoff <- 0
colnames(game_reg21_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"away_id", "away_name", "away_short", "sr_id_away", "reference_away",
"playoff")


##2021-22 postseason
game_pst21 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post21.json")
game_pst21_venue <- as.data.frame(game_pst21$games$venue)
game_pst21_venue[, 11] <- c(1:105)
colnames(game_pst21_venue)[11] <- "id_2"
game_pst21_home <- as.data.frame(game_pst21$games$home)
game_pst21_home[, 6] <- c(1:105)
colnames(game_pst21_home)[6] <- "id_2"
game_pst21_away <- as.data.frame(game_pst21$games$away)
game_pst21_away[, 6] <- c(1:105)
colnames(game_pst21_away)[6] <- "id_2"
game_pst21_game <- as.data.frame(game_pst21$games)
game_pst21_game <- game_pst21_game[, -c(10:13)]
game_pst21_game[, 10] <- c(1:105)
colnames(game_pst21_game)[10] <- "id_2"
game_pst21_all <- merge(game_pst21_game, game_pst21_venue, by =
"id_2")
game_pst21_all <- merge(game_pst21_all, game_pst21_home, by = "id_2")
game_pst21_all <- merge(game_pst21_all, game_pst21_away, by = "id_2")
game_pst21_all$playoff <- 1
colnames(game_pst21_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"away_id", "away_name", "away_short", "sr_id_away", "reference_away",
"playoff")

##combine 2021-22 regular and postseason
game21 <- rbind.fill(game_reg21_all, game_pst21_all)
##write to .csv
```

```
write.csv(game21, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game21.csv")

##data manipulation
game21_SQL <- game21
game21_SQL$home_short <- gsub("SJ", "SJS", game21_SQL$home_short)
game21_SQL$away_short <- gsub("SJ", "SJS", game21_SQL$away_short)
game21_SQL$home_short <- gsub("TB", "TBL", game21_SQL$home_short)
game21_SQL$away_short <- gsub("TB", "TBL", game21_SQL$away_short)
game21_SQL$home_short <- gsub("LA", "LAK", game21_SQL$home_short)
game21_SQL$away_short <- gsub("LA", "LAK", game21_SQL$away_short)
game21_SQL$home_short <- gsub("NJ", "NJD", game21_SQL$home_short)
game21_SQL$away_short <- gsub("NJ", "NJD", game21_SQL$away_short)
game21_SQL$home_short <- gsub("FLAK", "FLA", game21_SQL$home_short)
game21_SQL$away_short <- gsub("FLAK", "FLA", game21_SQL$away_short)
game21_SQL$winner <- ifelse(game21_SQL$home_points >
game21_SQL$away_points, game21_SQL$home_short,
ifelse(game21_SQL$home_points == game21_SQL$away_points, "tie",
game21_SQL$away_short))
game21_SQL$winner_score <- ifelse(game21_SQL$winner ==
game21_SQL$home_short, game21_SQL$home_points,
ifelse(game21_SQL$winner == "tie", game21_SQL$home_points,
game21_SQL$away_points))
game21_SQL$rivalry <- ifelse(game21_SQL$winner ==
game21_SQL$home_short, game21_SQL$away_short,
ifelse(game21_SQL$winner == "tie", "tie", game21_SQL$home_short))
game21_SQL$rivalry_score <- ifelse(game21_SQL$winner ==
game21_SQL$home_short, game21_SQL$away_points,
ifelse(game21_SQL$winner == "tie", game21_SQL$home_points,
game21_SQL$home_points))
game21_SQL_1 <- subset(game21_SQL, home_points != 0)
#game22_SQL_1$playoff[grepl("all-star",
tolower(game22_SQL_1$game_title))] <- 0
game21_SQL_1 <- subset(game21_SQL_1, !grepl("all-star",
tolower(game_title)))
game21_SQL_2 <- subset(game21_SQL_1, select = -c(1:5, 7:9, 11:22,
24:27, 29:30))
game21_SQL_2 <- game21_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game21_SQL_2 <- game21_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game21_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game21_cleaned.csv")
```

```
##2020-21 regular
game_reg20 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg20.json")
game_reg20_venue <- as.data.frame(game_reg20$games$venue)
game_reg20_venue[, 11] <- c(1:928)
colnames(game_reg20_venue)[11] <- "id_2"
game_reg20_home <- as.data.frame(game_reg20$games$home)
game_reg20_home[, 6] <- c(1:928)
colnames(game_reg20_home)[6] <- "id_2"
game_reg20_away <- as.data.frame(game_reg20$games$away)
game_reg20_away[, 6] <- c(1:928)
colnames(game_reg20_away)[6] <- "id_2"
game_reg20_game <- as.data.frame(game_reg20$games)
game_reg20_game <- game_reg20_game[, -c(9:12)]
game_reg20_game[, 9] <- c(1:928)
colnames(game_reg20_game)[9] <- "id_2"
game_reg20_all <- merge(game_reg20_game, game_reg20_venue, by =
"id_2")
game_reg20_all <- merge(game_reg20_all, game_reg20_home, by = "id_2")
game_reg20_all <- merge(game_reg20_all, game_reg20_away, by = "id_2")
game_reg20_all$playoff <- 0
colnames(game_reg20_all) <- c("id_2", "game_id", "status",
"coverage", "scheduled", "home_points", "away_points", "sr_id_game",
"reference_game", "venue_id", "venue_name", "venue_capacity",
"venue_address", "venue_city", "venue_state", "venue_zipcode",
"venue_country", "venue_timezone", "sr_id_venue", "home_id",
"home_name", "home_short", "sr_id_home", "reference_home", "away_id",
"away_name", "away_short", "sr_id_away", "reference_away", "playoff")

##2020-21 postseason
game_pst20 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post20.json")
game_pst20_venue <- as.data.frame(game_pst20$games$venue)
game_pst20_venue[, 11] <- c(1:105)
colnames(game_pst20_venue)[11] <- "id_2"
game_pst20_home <- as.data.frame(game_pst20$games$home)
game_pst20_home[, 7] <- c(1:105)
colnames(game_pst20_home)[7] <- "id_2"
game_pst20_away <- as.data.frame(game_pst20$games$away)
game_pst20_away[, 7] <- c(1:105)
colnames(game_pst20_away)[7] <- "id_2"
```

```
game_pst20_game <- as.data.frame(game_pst20$games)
game_pst20_game <- game_pst20_game[, -c(10:13)]
game_pst20_game[, 10] <- c(1:105)
colnames(game_pst20_game)[10] <- "id_2"
game_pst20_all <- merge(game_pst20_game, game_pst20_venue, by =
"id_2")
game_pst20_all <- merge(game_pst20_all, game_pst20_home, by = "id_2")
game_pst20_all <- merge(game_pst20_all, game_pst20_away, by = "id_2")
game_pst20_all <- game_pst20_all%>% relocate(seed.x, .after =
reference.y)
game_pst20_all <- game_pst20_all%>% relocate(seed.y, .after =
reference)
game_pst20_all$playoff <- 1
colnames(game_pst20_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"seed_home", "away_id", "away_name", "away_short", "sr_id_away",
"reference_away", "seed_away", "playoff")

##combine 2020-21 regular and postseason
game20 <- rbind.fill(game_reg20_all, game_pst20_all)
game20 <- game20 %>% relocate(game_title, .before = status)
##write to .csv
write.csv(game20, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game20.csv")

##data manipulation
game20_SQL <- game20
game20_SQL$home_short <- gsub("SJ", "SJS", game20_SQL$home_short)
game20_SQL$away_short <- gsub("SJ", "SJS", game20_SQL$away_short)
game20_SQL$home_short <- gsub("TB", "TBL", game20_SQL$home_short)
game20_SQL$away_short <- gsub("TB", "TBL", game20_SQL$away_short)
game20_SQL$home_short <- gsub("LA", "LAK", game20_SQL$home_short)
game20_SQL$away_short <- gsub("LA", "LAK", game20_SQL$away_short)
game20_SQL$home_short <- gsub("NJ", "NJD", game20_SQL$home_short)
game20_SQL$away_short <- gsub("NJ", "NJD", game20_SQL$away_short)
game20_SQL$home_short <- gsub("FLAK", "FLA", game20_SQL$home_short)
game20_SQL$away_short <- gsub("FLAK", "FLA", game20_SQL$away_short)
game20_SQL$winner <- ifelse(game20_SQL$home_points >
game20_SQL$away_points, game20_SQL$home_short,
```

```
ifelse(game20_SQL$home_points == game20_SQL$away_points, "tie",
game20_SQL$away_short))
game20_SQL$winner_score <- ifelse(game20_SQL$winner ==
game20_SQL$home_short, game20_SQL$home_points,
ifelse(game20_SQL$winner == "tie", game20_SQL$home_points,
game20_SQL$away_points))
game20_SQL$rivalry <- ifelse(game20_SQL$winner ==
game20_SQL$home_short, game20_SQL$away_short,
ifelse(game20_SQL$winner == "tie", "tie", game20_SQL$home_short))
game20_SQL$rivalry_score <- ifelse(game20_SQL$winner ==
game20_SQL$home_short, game20_SQL$away_points,
ifelse(game20_SQL$winner == "tie", game20_SQL$home_points,
game20_SQL$home_points))
game20_SQL_1 <- subset(game20_SQL, home_points != 0)
#game22_SQL_1$playoff[grepl("all-star",
tolower(game22_SQL_1$game_title))] <- 0
game20_SQL_1 <- subset(game20_SQL_1, !grepl("all-star",
tolower(game_title)))
game20_SQL_2 <- subset(game20_SQL_1, select = -c(1:5, 7:9, 11:22,
24:27, 29:30, 32:33))
game20_SQL_2 <- game20_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game20_SQL_2 <- game20_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game20_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game20_cleaned.csv")


##2019-20 regular
game_reg19 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg19.json")
game_reg19_venue <- as.data.frame(game_reg19$games$venue)
game_reg19_venue[, 11] <- c(1:1275)
colnames(game_reg19_venue)[11] <- "id_2"
game_reg19_home <- as.data.frame(game_reg19$games$home)
game_reg19_home[, 6] <- c(1:1275)
colnames(game_reg19_home)[6] <- "id_2"
game_reg19_away <- as.data.frame(game_reg19$games$away)
game_reg19_away[, 6] <- c(1:1275)
colnames(game_reg19_away)[6] <- "id_2"
game_reg19_game <- as.data.frame(game_reg19$games)
game_reg19_game <- game_reg19_game[, -c(9:12)]
game_reg19_game[, 10] <- c(1:1275)
```

```
colnames(game_reg19_game)[10] <- "id_2"
game_reg19_all <- merge(game_reg19_game, game_reg19_venue, by =
"id_2")
game_reg19_all <- merge(game_reg19_all, game_reg19_home, by = "id_2")
game_reg19_all <- merge(game_reg19_all, game_reg19_away, by = "id_2")
game_reg19_all <- game_reg19_all %>% relocate(title, .before =
status)
game_reg19_all <- game_reg19_all %>% relocate(zip, .before = country)
game_reg19_all$playoff <- 0
colnames(game_reg19_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"away_id", "away_name", "away_short", "sr_id_away", "reference_away",
"playoff")

##2019-20 postseason
game_pst19 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post19.json")
game_pst19_venue <- as.data.frame(game_pst19$games$venue)
game_pst19_venue[, 10] <- c(1:162)
colnames(game_pst19_venue)[10] <- "id_2"
game_pst19_home <- as.data.frame(game_pst19$games$home)
game_pst19_home[, 7] <- c(1:162)
colnames(game_pst19_home)[7] <- "id_2"
game_pst19_away <- as.data.frame(game_pst19$games$away)
game_pst19_away[, 7] <- c(1:162)
colnames(game_pst19_away)[7] <- "id_2"
game_pst19_game <- as.data.frame(game_pst19$games)
game_pst19_game <- game_pst19_game[, -c(10:13)]
game_pst19_game[, 10] <- c(1:162)
colnames(game_pst19_game)[10] <- "id_2"
game_pst19_all <- merge(game_pst19_game, game_pst19_venue, by =
"id_2")
game_pst19_all <- merge(game_pst19_all, game_pst19_home, by = "id_2")
game_pst19_all <- merge(game_pst19_all, game_pst19_away, by = "id_2")
game_pst19_all <- game_pst19_all%>% relocate(seed.x, .after =
reference.y)
game_pst19_all <- game_pst19_all%>% relocate(seed.y, .after =
reference)
game_pst19_all$playoff <- 1
```

```
colnames(game_pst19_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_country", "venue_timezone", "sr_id_venue", "home_id",
"home_name", "home_short", "sr_id_home", "reference_home",
"seed_home", "away_id", "away_name", "away_short", "sr_id_away",
"reference_away", "seed_away", "playoff")

##combine 2019-20 regular and postseason
game19 <- rbind.fill(game_reg19_all, game_pst19_all)
##write to .csv
write.csv(game19, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game19.csv")

##data manipulation
game19_SQL <- game19
game19_SQL$home_short <- gsub("SJ", "SJS", game19_SQL$home_short)
game19_SQL$away_short <- gsub("SJ", "SJS", game19_SQL$away_short)
game19_SQL$home_short <- gsub("TB", "TBL", game19_SQL$home_short)
game19_SQL$away_short <- gsub("TB", "TBL", game19_SQL$away_short)
game19_SQL$home_short <- gsub("LA", "LAK", game19_SQL$home_short)
game19_SQL$away_short <- gsub("LA", "LAK", game19_SQL$away_short)
game19_SQL$home_short <- gsub("NJ", "NJD", game19_SQL$home_short)
game19_SQL$away_short <- gsub("NJ", "NJD", game19_SQL$away_short)
game19_SQL$home_short <- gsub("FLAK", "FLA", game19_SQL$home_short)
game19_SQL$away_short <- gsub("FLAK", "FLA", game19_SQL$away_short)
game19_SQL$winner <- ifelse(game19_SQL$home_points >
game19_SQL$away_points, game19_SQL$home_short,
ifelse(game19_SQL$home_points == game19_SQL$away_points, "tie",
game19_SQL$away_short))
game19_SQL$winner_score <- ifelse(game19_SQL$winner ==
game19_SQL$home_short, game19_SQL$home_points,
ifelse(game19_SQL$winner == "tie", game19_SQL$home_points,
game19_SQL$away_points))
game19_SQL$rivalry <- ifelse(game19_SQL$winner ==
game19_SQL$home_short, game19_SQL$away_short,
ifelse(game19_SQL$winner == "tie", "tie", game19_SQL$home_short))
game19_SQL$rivalry_score <- ifelse(game19_SQL$winner ==
game19_SQL$home_short, game19_SQL$away_points,
ifelse(game19_SQL$winner == "tie", game19_SQL$home_points,
game19_SQL$home_points))
game19_SQL_1 <- subset(game19_SQL, home_points != 0)
```

```
game19_SQL_1 <- subset(game19_SQL_1, !grepl("all-star",
tolower(game_title)))
game19_SQL_2 <- subset(game19_SQL_1, select = -c(1:5, 7:9, 11:22,
24:27, 29:30, 32:33))
game19_SQL_2 <- game19_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game19_SQL_2 <- game19_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game19_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game19_cleaned.csv")


##2018-19 regular
game_reg18 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg18.json")
game_reg18_venue <- as.data.frame(game_reg18$games$venue)
game_reg18_venue[, 11] <- c(1:1274)
colnames(game_reg18_venue)[11] <- "id_2"
game_reg18_home <- as.data.frame(game_reg18$games$home)
game_reg18_home[, 6] <- c(1:1274)
colnames(game_reg18_home)[6] <- "id_2"
game_reg18_away <- as.data.frame(game_reg18$games$away)
game_reg18_away[, 6] <- c(1:1274)
colnames(game_reg18_away)[6] <- "id_2"
game_reg18_game <- as.data.frame(game_reg18$games)
game_reg18_game <- game_reg18_game[, -c(9:12)]
game_reg18_game[, 10] <- c(1:1274)
colnames(game_reg18_game)[10] <- "id_2"
game_reg18_all <- merge(game_reg18_game, game_reg18_venue, by =
"id_2")
game_reg18_all <- merge(game_reg18_all, game_reg18_home, by = "id_2")
game_reg18_all <- merge(game_reg18_all, game_reg18_away, by = "id_2")
game_reg18_all <- game_reg18_all %>% relocate(title, .before =
status)
game_reg18_all <- game_reg18_all %>% relocate(zip, .before = country)
game_reg18_all$playoff <- 0
colnames(game_reg18_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
```

```
"away_id", "away_name", "away_short", "sr_id_away", "reference_away",
"playoff")

##2018-19 postseason
game_pst18 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post18.json")
game_pst18_venue <- as.data.frame(game_pst18$games$venue)
game_pst18_venue[, 11] <- c(1:105)
colnames(game_pst18_venue)[11] <- "id_2"
game_pst18_home <- as.data.frame(game_pst18$games$home)
game_pst18_home[, 7] <- c(1:105)
colnames(game_pst18_home)[7] <- "id_2"
game_pst18_away <- as.data.frame(game_pst18$games$away)
game_pst18_away[, 7] <- c(1:105)
colnames(game_pst18_away)[7] <- "id_2"
game_pst18_game <- as.data.frame(game_pst18$games)
game_pst18_game <- game_pst18_game[, -c(10:13)]
game_pst18_game[, 10] <- c(1:105)
colnames(game_pst18_game)[10] <- "id_2"
game_pst18_all <- merge(game_pst18_game, game_pst18_venue, by =
"id_2")
game_pst18_all <- merge(game_pst18_all, game_pst18_home, by = "id_2")
game_pst18_all <- merge(game_pst18_all, game_pst18_away, by = "id_2")
game_pst18_all <- game_pst18_all%>% relocate(seed.x, .after =
reference.y)
game_pst18_all <- game_pst18_all%>% relocate(seed.y, .after =
reference)
game_pst18_all$playoff <- 1
colnames(game_pst18_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"seed_home", "away_id", "away_name", "away_short", "sr_id_away",
"reference_away", "seed_away", "playoff")

##combine 2018-19 regular and postseason
game18 <- rbind.fill(game_reg18_all, game_pst18_all)
##write to .csv
write.csv(game18, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game18.csv")
```

```
##data manipulation
game18_SQL <- game18
game18_SQL$home_short <- gsub("SJ", "SJS", game18_SQL$home_short)
game18_SQL$away_short <- gsub("SJ", "SJS", game18_SQL$away_short)
game18_SQL$home_short <- gsub("TB", "TBL", game18_SQL$home_short)
game18_SQL$away_short <- gsub("TB", "TBL", game18_SQL$away_short)
game18_SQL$home_short <- gsub("LA", "LAK", game18_SQL$home_short)
game18_SQL$away_short <- gsub("LA", "LAK", game18_SQL$away_short)
game18_SQL$home_short <- gsub("NJ", "NJD", game18_SQL$home_short)
game18_SQL$away_short <- gsub("NJ", "NJD", game18_SQL$away_short)
game18_SQL$home_short <- gsub("FLAK", "FLA", game18_SQL$home_short)
game18_SQL$away_short <- gsub("FLAK", "FLA", game18_SQL$away_short)
game18_SQL$winner <- ifelse(game18_SQL$home_points >
game18_SQL$away_points, game18_SQL$home_short,
ifelse(game18_SQL$home_points == game18_SQL$away_points, "tie",
game18_SQL$away_short))
game18_SQL$winner_score <- ifelse(game18_SQL$winner ==
game18_SQL$home_short, game18_SQL$home_points,
ifelse(game18_SQL$winner == "tie", game18_SQL$home_points,
game18_SQL$away_points))
game18_SQL$rivalry <- ifelse(game18_SQL$winner ==
game18_SQL$home_short, game18_SQL$away_short,
ifelse(game18_SQL$winner == "tie", "tie", game18_SQL$home_short))
game18_SQL$rivalry_score <- ifelse(game18_SQL$winner ==
game18_SQL$home_short, game18_SQL$away_points,
ifelse(game18_SQL$winner == "tie", game18_SQL$home_points,
game18_SQL$home_points))
game18_SQL_1 <- subset(game18_SQL, home_points != 0)
game18_SQL_1 <- subset(game18_SQL_1, !grepl("all-star",
tolower(game_title)))
game18_SQL_2 <- subset(game18_SQL_1, select = -c(1:5, 7:9, 11:22,
24:27, 29:30, 32:33))
game18_SQL_2 <- game18_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game18_SQL_2 <- game18_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game18_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game18_cleaned.csv")


##2017-18 regular
game_reg17 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg17.json")
```

```
game_reg17_venue <- as.data.frame(game_reg17$games$venue)
game_reg17_venue[, 11] <- c(1:1275)
colnames(game_reg17_venue)[11] <- "id_2"
game_reg17_home <- as.data.frame(game_reg17$games$home)
game_reg17_home[, 6] <- c(1:1275)
colnames(game_reg17_home)[6] <- "id_2"
game_reg17_away <- as.data.frame(game_reg17$games$away)
game_reg17_away[, 6] <- c(1:1275)
colnames(game_reg17_away)[6] <- "id_2"
game_reg17_game <- as.data.frame(game_reg17$games)
game_reg17_game <- game_reg17_game[, -c(9:11)]
game_reg17_game[, 10] <- c(1:1275)
colnames(game_reg17_game)[10] <- "id_2"
game_reg17_all <- merge(game_reg17_game, game_reg17_venue, by =
"id_2")
game_reg17_all <- merge(game_reg17_all, game_reg17_home, by = "id_2")
game_reg17_all <- merge(game_reg17_all, game_reg17_away, by = "id_2")
game_reg17_all <- game_reg17_all %>% relocate(title, .before =
status)
game_reg17_all$playoff <- 0
colnames(game_reg17_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"away_id", "away_name", "away_short", "sr_id_away", "reference_away",
"playoff")

##2017-18 postseason
game_pst17 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post17.json")
game_pst17_venue <- as.data.frame(game_pst17$games$venue)
game_pst17_venue[, 11] <- c(1:105)
colnames(game_pst17_venue)[11] <- "id_2"
game_pst17_home <- as.data.frame(game_pst17$games$home)
game_pst17_home[, 7] <- c(1:105)
colnames(game_pst17_home)[7] <- "id_2"
game_pst17_away <- as.data.frame(game_pst17$games$away)
game_pst17_away[, 7] <- c(1:105)
colnames(game_pst17_away)[7] <- "id_2"
game_pst17_game <- as.data.frame(game_pst17$games)
game_pst17_game <- game_pst17_game[, -c(10:12)]
game_pst17_game[, 10] <- c(1:105)
```

```
colnames(game_pst17_game)[10] <- "id_2"
game_pst17_all <- merge(game_pst17_game, game_pst17_venue, by =
"id_2")
game_pst17_all <- merge(game_pst17_all, game_pst17_home, by = "id_2")
game_pst17_all <- merge(game_pst17_all, game_pst17_away, by = "id_2")
game_pst17_all <- game_pst17_all%>% relocate(seed.x, .after =
reference.y)
game_pst17_all <- game_pst17_all%>% relocate(seed.y, .after =
reference)
game_pst17_all$playoff <- 1
colnames(game_pst17_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "reference_home",
"seed_home", "away_id", "away_name", "away_short", "sr_id_away",
"reference_away", "seed_away", "playoff")

##combine 2017-18regular and postseason
game17 <- rbind.fill(game_reg17_all, game_pst17_all)
##write to .csv
write.csv(game17, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game17.csv")

##data manipulation
game17_SQL <- game17
game17_SQL$home_short <- gsub("SJ", "SJS", game17_SQL$home_short)
game17_SQL$away_short <- gsub("SJ", "SJS", game17_SQL$away_short)
game17_SQL$home_short <- gsub("TB", "TBL", game17_SQL$home_short)
game17_SQL$away_short <- gsub("TB", "TBL", game17_SQL$away_short)
game17_SQL$home_short <- gsub("LA", "LAK", game17_SQL$home_short)
game17_SQL$away_short <- gsub("LA", "LAK", game17_SQL$away_short)
game17_SQL$home_short <- gsub("NJ", "NJD", game17_SQL$home_short)
game17_SQL$away_short <- gsub("NJ", "NJD", game17_SQL$away_short)
game17_SQL$home_short <- gsub("FLAK", "FLA", game17_SQL$home_short)
game17_SQL$away_short <- gsub("FLAK", "FLA", game17_SQL$away_short)
game17_SQL$winner <- ifelse(game17_SQL$home_points >
game17_SQL$away_points, game17_SQL$home_short,
ifelse(game17_SQL$home_points == game17_SQL$away_points, "tie",
game17_SQL$away_short))
game17_SQL$winner_score <- ifelse(game17_SQL$winner ==
game17_SQL$home_short, game17_SQL$home_points,
```

```
ifelse(game17_SQL$winner == "tie", game17_SQL$home_points,
game17_SQL$away_points))
game17_SQL$rivalry <- ifelse(game17_SQL$winner ==
game17_SQL$home_short, game17_SQL$away_short,
ifelse(game17_SQL$winner == "tie", "tie", game17_SQL$home_short))
game17_SQL$rivalry_score <- ifelse(game17_SQL$winner ==
game17_SQL$home_short, game17_SQL$away_points,
ifelse(game17_SQL$winner == "tie", game17_SQL$home_points,
game17_SQL$home_points))
game17_SQL_1 <- subset(game17_SQL, home_points != 0)
game17_SQL_1 <- subset(game17_SQL_1, !grepl("all-star",
tolower(game_title)))
game17_SQL_2 <- subset(game17_SQL_1, select = -c(1:5, 7:9, 11:22,
24:27, 29:30, 32:33))
game17_SQL_2 <- game17_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game17_SQL_2 <- game17_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game17_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game17_cleaned.csv")

##2016-17 regular
game_reg16 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg16.json")
game_reg16_venue <- as.data.frame(game_reg16$games$venue)
game_reg16_venue[, 11] <- c(1:1235)
colnames(game_reg16_venue)[11] <- "id_2"
game_reg16_home <- as.data.frame(game_reg16$games$home)
game_reg16_home[, 5] <- c(1:1235)
colnames(game_reg16_home)[5] <- "id_2"
game_reg16_away <- as.data.frame(game_reg16$games$away)
game_reg16_away[, 5] <- c(1:1235)
colnames(game_reg16_away)[5] <- "id_2"
game_reg16_game <- as.data.frame(game_reg16$games)
game_reg16_game <- game_reg16_game[, -c(9:12)]
game_reg16_game[, 10] <- c(1:1235)
colnames(game_reg16_game)[10] <- "id_2"
game_reg16_all <- merge(game_reg16_game, game_reg16_venue, by =
"id_2")
game_reg16_all <- merge(game_reg16_all, game_reg16_home, by = "id_2")
game_reg16_all <- merge(game_reg16_all, game_reg16_away, by = "id_2")
game_reg16_all <- game_reg16_all %>% relocate(title, .before =
status)
```

```
game_reg16_all <- game_reg16_all %>% relocate(zip, .before = country)
game_reg16_all$playoff <- 0
colnames(game_reg16_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "sr_id_home", "away_id",
"away_name", "away_short", "sr_id_away", "playoff")

##2016-17 postseason
game_pst16 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post16.json")
game_pst16_venue <- as.data.frame(game_pst16$games$venue)
game_pst16_venue[, 11] <- c(1:105)
colnames(game_pst16_venue)[11] <- "id_2"
game_pst16_home <- as.data.frame(game_pst16$games$home)
game_pst16_home[, 6] <- c(1:105)
colnames(game_pst16_home)[6] <- "id_2"
game_pst16_away <- as.data.frame(game_pst16$games$away)
game_pst16_away[, 6] <- c(1:105)
colnames(game_pst16_away)[6] <- "id_2"
game_pst16_game <- as.data.frame(game_pst16$games)
game_pst16_game <- game_pst16_game[, -c(10:12)]
game_pst16_game[, 10] <- c(1:105)
colnames(game_pst16_game)[10] <- "id_2"
game_pst16_all <- merge(game_pst16_game, game_pst16_venue, by =
"id_2")
game_pst16_all <- merge(game_pst16_all, game_pst16_home, by = "id_2")
game_pst16_all <- merge(game_pst16_all, game_pst16_away, by = "id_2")
game_pst16_all$playoff <- 1
colnames(game_pst16_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "home_points", "away_points",
"sr_id_game", "reference_game", "venue_id", "venue_name",
"venue_capacity", "venue_address", "venue_city", "venue_state",
"venue_zipcode", "venue_country", "venue_timezone", "sr_id_venue",
"home_id", "home_name", "home_short", "seed_home", "sr_id_home",
"away_id", "away_name", "away_short", "seed_away", "sr_id_away",
"playoff")

##combine 2016-17 regular and postseason
game16 <- rbind.fill(game_reg16_all, game_pst16_all)
##write to .csv
```

```
write.csv(game16, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game16.csv")

##data manipulation
game16_SQL <- game16
game16_SQL$home_short <- gsub("SJ", "SJS", game16_SQL$home_short)
game16_SQL$away_short <- gsub("SJ", "SJS", game16_SQL$away_short)
game16_SQL$home_short <- gsub("TB", "TBL", game16_SQL$home_short)
game16_SQL$away_short <- gsub("TB", "TBL", game16_SQL$away_short)
game16_SQL$home_short <- gsub("LA", "LAK", game16_SQL$home_short)
game16_SQL$away_short <- gsub("LA", "LAK", game16_SQL$away_short)
game16_SQL$home_short <- gsub("NJ", "NJD", game16_SQL$home_short)
game16_SQL$away_short <- gsub("NJ", "NJD", game16_SQL$away_short)
game16_SQL$home_short <- gsub("FLAK", "FLA", game16_SQL$home_short)
game16_SQL$away_short <- gsub("FLAK", "FLA", game16_SQL$away_short)
game16_SQL$winner <- ifelse(game16_SQL$home_points >
game16_SQL$away_points, game16_SQL$home_short,
ifelse(game16_SQL$home_points == game16_SQL$away_points, "tie",
game16_SQL$away_short))
game16_SQL$winner_score <- ifelse(game16_SQL$winner ==
game16_SQL$home_short, game16_SQL$home_points,
ifelse(game16_SQL$winner == "tie", game16_SQL$home_points,
game16_SQL$away_points))
game16_SQL$rivalry <- ifelse(game16_SQL$winner ==
game16_SQL$home_short, game16_SQL$away_short,
ifelse(game16_SQL$winner == "tie", "tie", game16_SQL$home_short))
game16_SQL$rivalry_score <- ifelse(game16_SQL$winner ==
game16_SQL$home_short, game16_SQL$away_points,
ifelse(game16_SQL$winner == "tie", game16_SQL$home_points,
game16_SQL$home_points))
game16_SQL_1 <- subset(game16_SQL, home_points != 0)
game16_SQL_1 <- subset(game16_SQL_1, !grepl("all-star",
tolower(game_title)))
game16_SQL_2 <- subset(game16_SQL_1, select = -c(1:5, 7:9, 11:22,
24:26, 28, 30:31))
game16_SQL_2 <- game16_SQL_2 %>% relocate(reference_game, .before =
scheduled)
game16_SQL_2 <- game16_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game16_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game16_cleaned.csv")
```

```
##2015-16 regular
game_reg15 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg15.json")
game_reg15_venue <- as.data.frame(game_reg15$games$venue)
game_reg15_venue[, 11] <- c(1:1236)
colnames(game_reg15_venue)[11] <- "id_2"
game_reg15_home <- as.data.frame(game_reg15$games$home)
game_reg15_home[, 5] <- c(1:1236)
colnames(game_reg15_home)[5] <- "id_2"
game_reg15_away <- as.data.frame(game_reg15$games$away)
game_reg15_away[, 5] <- c(1:1236)
colnames(game_reg15_away)[5] <- "id_2"
game_reg15_game <- as.data.frame(game_reg15$games)
game_reg15_game <- game_reg15_game[, -c(6:9)]
game_reg15_game[, 9] <- c(1:1236)
colnames(game_reg15_game)[9] <- "id_2"
game_reg15_all <- merge(game_reg15_game, game_reg15_venue, by =
"id_2")
game_reg15_all <- merge(game_reg15_all, game_reg15_home, by = "id_2")
game_reg15_all <- merge(game_reg15_all, game_reg15_away, by = "id_2")
game_reg15_all <- game_reg15_all %>% relocate(title, .before =
status)
game_reg15_all <- game_reg15_all %>% relocate(zip, .before = country)
game_reg15_all$playoff <- 0
colnames(game_reg15_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled",  "sr_id_game", "home_points",
"away_points", "venue_id", "venue_name", "venue_capacity",
"venue_address", "venue_city", "venue_state", "venue_zipcode",
"venue_country", "venue_timezone", "sr_id_venue", "home_id",
"home_name", "home_short", "sr_id_home", "away_id", "away_name",
"away_short", "sr_id_away", "playoff")

##2015-16 postseason
game_pst15 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post15.json")
game_pst15_venue <- as.data.frame(game_pst15$games$venue)
game_pst15_venue[, 11] <- c(1:105)
colnames(game_pst15_venue)[11] <- "id_2"
game_pst15_home <- as.data.frame(game_pst15$games$home)
game_pst15_home[, 6] <- c(1:105)
colnames(game_pst15_home)[6] <- "id_2"
game_pst15_away <- as.data.frame(game_pst15$games$away)
game_pst15_away[, 6] <- c(1:105)
```

```
colnames(game_pst15_away)[6] <- "id_2"
game_pst15_game <- as.data.frame(game_pst15$games)
game_pst15_game <- game_pst15_game[, -c(7:9)]
game_pst15_game[, 7] <- c(1:105)
colnames(game_pst15_game)[7] <- "id_2"
game_pst15_all <- merge(game_pst15_game, game_pst15_venue, by =
"id_2")
game_pst15_all <- merge(game_pst15_all, game_pst15_home, by = "id_2")
game_pst15_all <- merge(game_pst15_all, game_pst15_away, by = "id_2")
game_pst15_all$playoff <- 1
colnames(game_pst15_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "sr_id_game", "venue_id",
"venue_name", "venue_capacity", "venue_address", "venue_city",
"venue_state", "venue_zipcode", "venue_country", "venue_timezone",
"sr_id_venue", "home_id", "home_name", "home_short", "seed_home",
"sr_id_home", "away_id", "away_name", "away_short", "seed_away",
"sr_id_away", "playoff")

##combine 2015-16 regular and postseason
game15 <- rbind.fill(game_reg15_all, game_pst15_all)
##write to .csv
write.csv(game15, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game15.csv")

##data manipulation
game15_SQL <- game15
game15_SQL$home_short <- gsub("SJ", "SJS", game15_SQL$home_short)
game15_SQL$away_short <- gsub("SJ", "SJS", game15_SQL$away_short)
game15_SQL$home_short <- gsub("TB", "TBL", game15_SQL$home_short)
game15_SQL$away_short <- gsub("TB", "TBL", game15_SQL$away_short)
game15_SQL$home_short <- gsub("LA", "LAK", game15_SQL$home_short)
game15_SQL$away_short <- gsub("LA", "LAK", game15_SQL$away_short)
game15_SQL$home_short <- gsub("NJ", "NJD", game15_SQL$home_short)
game15_SQL$away_short <- gsub("NJ", "NJD", game15_SQL$away_short)
game15_SQL$home_short <- gsub("FLAK", "FLA", game15_SQL$home_short)
game15_SQL$away_short <- gsub("FLAK", "FLA", game15_SQL$away_short)
game15_SQL$winner <- ifelse(game15_SQL$home_points >
game15_SQL$away_points, game15_SQL$home_short,
ifelse(game15_SQL$home_points == game15_SQL$away_points, "tie",
game15_SQL$away_short))
game15_SQL$winner_score <- ifelse(game15_SQL$winner ==
game15_SQL$home_short, game15_SQL$home_points,
ifelse(game15_SQL$winner == "tie", game15_SQL$home_points,
game15_SQL$away_points))
```

```
game15_SQL$rivalry <- ifelse(game15_SQL$winner ==
game15_SQL$home_short, game15_SQL$away_short,
ifelse(game15_SQL$winner == "tie", "tie", game15_SQL$home_short))
game15_SQL$rivalry_score <- ifelse(game15_SQL$winner ==
game15_SQL$home_short, game15_SQL$away_points,
ifelse(game15_SQL$winner == "tie", game15_SQL$home_points,
game15_SQL$home_points))
game15_SQL_1 <- subset(game15_SQL, home_points != 0)
game15_SQL_1 <- subset(game15_SQL_1, !grepl("all-star",
tolower(game_title)))
game15_SQL_2 <- subset(game15_SQL_1, select = -c(1:5, 7:21, 23:25,
27, 29:30))
game15_SQL_2 <- game15_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game15_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game15_cleaned.csv")


##2014-15 regular
game_reg14 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg14.json")
game_reg14_venue <- as.data.frame(game_reg14$games$venue)
game_reg14_venue[, 11] <- c(1:1233)
colnames(game_reg14_venue)[11] <- "id_2"
game_reg14_home <- as.data.frame(game_reg14$games$home)
game_reg14_home[, 5] <- c(1:1233)
colnames(game_reg14_home)[5] <- "id_2"
game_reg14_away <- as.data.frame(game_reg14$games$away)
game_reg14_away[, 5] <- c(1:1233)
colnames(game_reg14_away)[5] <- "id_2"
game_reg14_game <- as.data.frame(game_reg14$games)
game_reg14_game <- game_reg14_game[, -c(6:9)]
game_reg14_game[, 9] <- c(1:1233)
colnames(game_reg14_game)[9] <- "id_2"
game_reg14_all <- merge(game_reg14_game, game_reg14_venue, by =
"id_2")
game_reg14_all <- merge(game_reg14_all, game_reg14_home, by = "id_2")
game_reg14_all <- merge(game_reg14_all, game_reg14_away, by = "id_2")
game_reg14_all <- game_reg14_all %>% relocate(title, .before =
status)
game_reg14_all <- game_reg14_all %>% relocate(zip, .before = country)
game_reg14_all$playoff <- 0
```

```
colnames(game_reg14_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled",  "sr_id_game", "home_points",
"away_points", "venue_id", "venue_name", "venue_capacity",
"venue_address", "venue_city", "venue_state", "venue_zipcode",
"venue_country", "venue_timezone", "sr_id_venue", "home_id",
"home_name", "home_short", "sr_id_home", "away_id", "away_name",
"away_short", "sr_id_away", "playoff")

##2014-15 postseason
game_pst14 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post14.json")
game_pst14_venue <- as.data.frame(game_pst14$games$venue)
game_pst14_venue[, 11] <- c(1:105)
colnames(game_pst14_venue)[11] <- "id_2"
game_pst14_home <- as.data.frame(game_pst14$games$home)
game_pst14_home[, 6] <- c(1:105)
colnames(game_pst14_home)[6] <- "id_2"
game_pst14_away <- as.data.frame(game_pst14$games$away)
game_pst14_away[, 6] <- c(1:105)
colnames(game_pst14_away)[6] <- "id_2"
game_pst14_game <- as.data.frame(game_pst14$games)
game_pst14_game <- game_pst14_game[, -c(7:10)]
game_pst14_game[, 9] <- c(1:105)
colnames(game_pst14_game)[9] <- "id_2"
game_pst14_all <- merge(game_pst14_game, game_pst14_venue, by =
"id_2")
game_pst14_all <- merge(game_pst14_all, game_pst14_home, by = "id_2")
game_pst14_all <- merge(game_pst14_all, game_pst14_away, by = "id_2")
game_pst14_all <- game_pst14_all %>% relocate(zip, .before = country)
game_pst14_all$playoff <- 1
colnames(game_pst14_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "sr_id_game", "home_points",
"away_points", "venue_id", "venue_name", "venue_capacity",
"venue_address", "venue_city", "venue_state", "venue_zipcode",
"venue_country", "venue_timezone", "sr_id_venue", "home_id",
"home_name", "home_short", "seed_home", "sr_id_home", "away_id",
"away_name", "away_short", "seed_away", "sr_id_away", "playoff")

##combine 2014-15 regular and postseason
game14 <- rbind.fill(game_reg14_all, game_pst14_all)
##write to .csv
write.csv(game14, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game14.csv")
```

```
##data manipulation
game14_SQL <- game14
game14_SQL$home_short <- gsub("SJ", "SJS", game14_SQL$home_short)
game14_SQL$away_short <- gsub("SJ", "SJS", game14_SQL$away_short)
game14_SQL$home_short <- gsub("TB", "TBL", game14_SQL$home_short)
game14_SQL$away_short <- gsub("TB", "TBL", game14_SQL$away_short)
game14_SQL$home_short <- gsub("LA", "LAK", game14_SQL$home_short)
game14_SQL$away_short <- gsub("LA", "LAK", game14_SQL$away_short)
game14_SQL$home_short <- gsub("NJ", "NJD", game14_SQL$home_short)
game14_SQL$away_short <- gsub("NJ", "NJD", game14_SQL$away_short)
game14_SQL$home_short <- gsub("FLAK", "FLA", game14_SQL$home_short)
game14_SQL$away_short <- gsub("FLAK", "FLA", game14_SQL$away_short)
game14_SQL$winner <- ifelse(game14_SQL$home_points >
game14_SQL$away_points, game14_SQL$home_short,
ifelse(game14_SQL$home_points == game14_SQL$away_points, "tie",
game14_SQL$away_short))
game14_SQL$winner_score <- ifelse(game14_SQL$winner ==
game14_SQL$home_short, game14_SQL$home_points,
ifelse(game14_SQL$winner == "tie", game14_SQL$home_points,
game14_SQL$away_points))
game14_SQL$rivalry <- ifelse(game14_SQL$winner ==
game14_SQL$home_short, game14_SQL$away_short,
ifelse(game14_SQL$winner == "tie", "tie", game14_SQL$home_short))
game14_SQL$rivalry_score <- ifelse(game14_SQL$winner ==
game14_SQL$home_short, game14_SQL$away_points,
ifelse(game14_SQL$winner == "tie", game14_SQL$home_points,
game14_SQL$home_points))
game14_SQL_1 <- subset(game14_SQL, home_points != 0)
game14_SQL_1 <- subset(game14_SQL_1, !grepl("all-star",
tolower(game_title)))
game14_SQL_2 <- subset(game14_SQL_1, select = -c(1:5, 7:21, 23:25,
27, 29:30))
game14_SQL_2 <- game14_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game14_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game14_cleaned.csv")


##2013-14 regular
game_reg13 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-reg13.json")
game_reg13_venue <- as.data.frame(game_reg13$games$venue)
```

```
game_reg13_venue[, 11] <- c(1:1233)
colnames(game_reg13_venue)[11] <- "id_2"
game_reg13_home <- as.data.frame(game_reg13$games$home)
game_reg13_home[, 5] <- c(1:1233)
colnames(game_reg13_home)[5] <- "id_2"
game_reg13_away <- as.data.frame(game_reg13$games$away)
game_reg13_away[, 5] <- c(1:1233)
colnames(game_reg13_away)[5] <- "id_2"
game_reg13_game <- as.data.frame(game_reg13$games)
game_reg13_game <- game_reg13_game[, -c(6:9)]
game_reg13_game[, 9] <- c(1:1233)
colnames(game_reg13_game)[9] <- "id_2"
game_reg13_all <- merge(game_reg13_game, game_reg13_venue, by =
"id_2")
game_reg13_all <- merge(game_reg13_all, game_reg13_home, by = "id_2")
game_reg13_all <- merge(game_reg13_all, game_reg13_away, by = "id_2")
game_reg13_all <- game_reg13_all %>% relocate(title, .before =
status)
game_reg13_all <- game_reg13_all %>% relocate(zip, .before = country)
game_reg13_all$playoff <- 0
colnames(game_reg13_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled",  "sr_id_game", "home_points",
"away_points", "venue_id", "venue_name", "venue_capacity",
"venue_address", "venue_city", "venue_state", "venue_zipcode",
"venue_country", "venue_timezone", "sr_id_venue", "home_id",
"home_name", "home_short", "sr_id_home", "away_id", "away_name",
"away_short", "sr_id_away", "playoff")

##2013-14 postseason
game_pst13 <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/schedule-post13.json")
game_pst13_venue <- as.data.frame(game_pst13$games$venue)
game_pst13_venue[, 11] <- c(1:105)
colnames(game_pst13_venue)[11] <- "id_2"
game_pst13_home <- as.data.frame(game_pst13$games$home)
game_pst13_home[, 6] <- c(1:105)
colnames(game_pst13_home)[6] <- "id_2"
game_pst13_away <- as.data.frame(game_pst13$games$away)
game_pst13_away[, 6] <- c(1:105)
colnames(game_pst13_away)[6] <- "id_2"
game_pst13_game <- as.data.frame(game_pst13$games)
game_pst13_game <- game_pst13_game[, -c(7:9)]
game_pst13_game[, 7] <- c(1:105)
colnames(game_pst13_game)[7] <- "id_2"
```

```
game_pst13_all <- merge(game_pst13_game, game_pst13_venue, by =
"id_2")
game_pst13_all <- merge(game_pst13_all, game_pst13_home, by = "id_2")
game_pst13_all <- merge(game_pst13_all, game_pst13_away, by = "id_2")
game_pst13_all <- game_pst13_all %>% relocate(zip, .before = country)
game_pst13_all$playoff <- 1
colnames(game_pst13_all) <- c("id_2", "game_id", "game_title",
"status", "coverage", "scheduled", "sr_id_game", "venue_id",
"venue_name", "venue_capacity", "venue_address", "venue_city",
"venue_state", "venue_zipcode", "venue_country", "venue_timezone",
"sr_id_venue", "home_id", "home_name", "home_short", "seed_home",
"sr_id_home", "away_id", "away_name", "away_short", "seed_away",
"sr_id_away", "playoff")

##combine 2013-14 regular and postseason
game13 <- rbind.fill(game_reg13_all, game_pst13_all)
##write to .csv
write.csv(game13, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game13.csv")

##data manipulation
game13_SQL <- game13
game13_SQL$home_short <- gsub("SJ", "SJS", game13_SQL$home_short)
game13_SQL$away_short <- gsub("SJ", "SJS", game13_SQL$away_short)
game13_SQL$home_short <- gsub("TB", "TBL", game13_SQL$home_short)
game13_SQL$away_short <- gsub("TB", "TBL", game13_SQL$away_short)
game13_SQL$home_short <- gsub("LA", "LAK", game13_SQL$home_short)
game13_SQL$away_short <- gsub("LA", "LAK", game13_SQL$away_short)
game13_SQL$home_short <- gsub("NJ", "NJD", game13_SQL$home_short)
game13_SQL$away_short <- gsub("NJ", "NJD", game13_SQL$away_short)
game13_SQL$home_short <- gsub("FLAK", "FLA", game13_SQL$home_short)
game13_SQL$away_short <- gsub("FLAK", "FLA", game13_SQL$away_short)
game13_SQL$winner <- ifelse(game13_SQL$home_points >
game13_SQL$away_points, game13_SQL$home_short,
ifelse(game13_SQL$home_points == game13_SQL$away_points, "tie",
game13_SQL$away_short))
game13_SQL$winner_score <- ifelse(game13_SQL$winner ==
game13_SQL$home_short, game13_SQL$home_points,
ifelse(game13_SQL$winner == "tie", game13_SQL$home_points,
game13_SQL$away_points))
game13_SQL$rivalry <- ifelse(game13_SQL$winner ==
game13_SQL$home_short, game13_SQL$away_short,
ifelse(game13_SQL$winner == "tie", "tie", game13_SQL$home_short))
```

```
game13_SQL$rivalry_score <- ifelse(game13_SQL$winner ==
game13_SQL$home_short, game13_SQL$away_points,
ifelse(game13_SQL$winner == "tie", game13_SQL$home_points,
game13_SQL$home_points))
game13_SQL_1 <- subset(game13_SQL, home_points != 0)
game13_SQL_1 <- subset(game13_SQL_1, !grepl("all-star",
tolower(game_title)))
game13_SQL_2 <- subset(game13_SQL_1, select = -c(1:5, 7:21, 23:25,
27, 29:30))
game13_SQL_2 <- game13_SQL_2 %>% relocate(playoff, .after =
rivalry_score)
write.csv(game13_SQL_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/game13_cleaned.csv")

##combine all rows from each season
game <- rbind.fill(game13_SQL_2, game14_SQL_2, game15_SQL_2,
game16_SQL_2, game17_SQL_2, game18_SQL_2, game19_SQL_2, game20_SQL_2,
game21_SQL_2, game22_SQL_2)
game <- game %>% relocate(reference_game, .before = scheduled)
rows_to_modify <- c(4884:4998)
game$scheduled[rows_to_modify] <-
substr(game$scheduled[rows_to_modify], 1,
nchar(game$scheduled[rows_to_modify])-5)
rows_to_modify_1 <- c(1197:2558)
game$scheduled[rows_to_modify_1] <-
substr(game$scheduled[rows_to_modify_1], 1,
nchar(game$scheduled[rows_to_modify_1])-5)
rows_to_modify_2 <- c(3759:3841)
game$scheduled[rows_to_modify_2] <-
substr(game$scheduled[rows_to_modify_2], 1,
nchar(game$scheduled[rows_to_modify_2])-5)
game$date <- str_sub(game$scheduled, 1, -11)
game <- game[, -2]
game <- game %>% relocate(date, .after = reference_game)
write.csv(game, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/game/0. game.csv")


# for player statistics table
player_stat_df <- data.frame()
page <- c(1:31)
year <- c(2011:2024)
for (x in year){
```

```
  for(i in page) {
    player_stat_url <-
paste0("https://www.capfriendly.com/browse/active/", x,
"/salary?stats-season=", x,
"&age-calculation-date=today&display=skater-individual-advanced-stats
,goalie-advanced-stats&hide=clauses,age,position,handed,expiry-status
,salary,caphit&pg=", i)
    # check if the url is valid
    #if (url.exists(salary_url)) {
    #  salary_url <- salary_url
    #}
    url_html <- read_html(player_stat_url)
    url_nodes <- html_nodes(url_html, "table")
    player_stat_table <- html_table(url_nodes)[[1]]
    player_stat <- player_stat_url %>%
      read_html() %>%
      html_nodes("table") %>%
      html_table()
    player_stat_df_1 <- player_stat[[1]] %>%
      mutate(year = x)
    player_stat_df <- rbind.fill(player_stat_df, player_stat_df_1)
  }
}

colnames(player_stat_df) <- c("player_name", "team", "games_played",
"goals", "assists", "points", "points_per_game", "plus/minus",
"shots_on_goal", "shooting_percentage", "average_time_on_ice",
"individual_expected_goals", "individual_shots_on_goal",
"individual_corsi", "individual_fenwick",
"individual_expected_goals_per60min",
"individual_shots_on_goal_per60min", "individual_corsi_per60min",
"individual_fenwick_per60min", "wins", "loses", "shutouts",
"goals_against_average", "save_percentage",
"goals_against_per60min_of_ice_time",
"goaltender_related_expected_goals_against_per60min_of_ice_time",
"goals_saved_above_expected_per60min_of_ice_time", "year")

player_stat_df_2 <- player_stat_df %>%
  separate(player_name, into = c("number", "first_name", "last_name",
"more_name", "more_name_2"), sep = " ", remove = TRUE) %>%
  mutate(player_name = paste(ifelse(!is.na(first_name), first_name,
""),
                             ifelse(!is.na(last_name), last_name,
""),
```

```
                              ifelse(!is.na(more_name), more_name,
""),
                              ifelse(!is.na(more_name_2), more_name_2,
""),
                              sep = " ")) %>%
  separate(average_time_on_ice, into = c("average_min_on_ice",
"average_sec_on_ice"), sep = ":", remove = TRUE) %>%
  relocate(player_name, .before = team)
player_stat_df_2 <- player_stat_df_2[, -c(1:5)]
player_stat_df_2$player_name <-
str_trim(player_stat_df_2$player_name, side = "right")

write.csv(player_stat_df_2, "G:/My Drive/0. study abroad/academic/10.
2023 Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/player stat/player_stat_scraped.csv")


# for salary table
#yearly_salary <-
read_html("https://www.spotrac.com/nhl/rankings/earnings/")
#nodes_yearly_salary <- html_nodes(yearly_salary, xpath = '//tbody')
#yearly_salary_tab <- html_table(nodes_yearly_salary)[[1]]
#colnames(yearly_salary_tab) <- c("No", "name_drafted",
"earnings_total", "seasons", "earnings_total_2")
salary_df <- data.frame()
page <- c(1:31)
year <- c(2011:2024)
for (x in year){
  for(i in page) {
    salary_url <-
paste0("https://www.capfriendly.com/browse/active/",
                           x,
                           "?stats-season=",
                           x,

"&age-calculation-date=today&display=draft,signing-age&hide=clauses,a
ge,position,handed,expiry-status,caphit&pg=",
                           i)
    # check if the url is valid
    #if (url.exists(salary_url)) {
    #  salary_url <- salary_url
    #}
    url_html <- read_html(salary_url)
    url_nodes <- html_nodes(url_html, "table")
    salary_table <- html_table(url_nodes)[[1]]
```

```
    salary <- salary_url %>%
      read_html() %>%
      html_nodes("table") %>%
      html_table()
    salary_df_1 <- salary[[1]] %>%
      mutate(year = x - 1)
    salary_df <- rbind.fill(salary_df, salary_df_1)
  }
}
head(salary_df)
# change column names
colnames(salary_df) <- c("player_name", "team", "drafted",
                         "games_played", "goals", "assists",
                         "points", "points_per_game",
                         "plus/minus", "shots_on_goal",
                         "shooting_percentage",
                         "average_time_on_ice", "wins",
                         "loses", "shootouts",
                         "goals against average",
                         "saving_percentage", "signing_age",
                         "salary", "year")

# data management
# cut "$" and "," in the salary column,
# convert the salary values to numeric,
# drop rows that have 0 for their salaries,
# and split the player_name column so that
# the numbers leading the names do not matter
salary_df_2 <- salary_df %>%
  mutate(salary = str_replace_all(salary, "\\$", "")) %>%
  mutate(salary = str_replace_all(salary, ",", "")) %>%
  filter(as.numeric(unlist(salary)) != 0) %>%
  separate(player_name,
           into = c("number", "first_name",
                    "last_name", "more_name",
                    "more_name_2"),
           sep = " ",
           remove = TRUE) %>%
  mutate(player_name = paste(ifelse(!is.na(first_name),
                                    first_name, ""),
                             ifelse(!is.na(last_name),
                                    last_name, ""),
                             ifelse(!is.na(more_name),
                                    more_name, ""),
                             ifelse(!is.na(more_name_2),
```

```
                                    more_name_2, ""),
                              sep = " ")) %>%
  relocate(player_name, .before = team) %>%
  mutate(drafted_year = substr(drafted,
                               nchar(drafted) - 9,
                               nchar(drafted) - 6)) %>%
  relocate(drafted_year, .after = drafted)
salary_df_2 <- salary_df_2[, -c(1:5, 8)]
salary_df_2$salary <- as.numeric(salary_df_2$salary)


# stanley cup
##yearly winners
stanley_cup <-
read_html("https://blog.ticketcity.com/nhl/stanley-cup-champions/")
nodes_stanley_cup <- html_nodes(stanley_cup, "table")
stanley_cup_table <- html_table(nodes_stanley_cup)[[1]]
colnames(stanley_cup_table) <- c("year", "winning_team",
"losing_team", "champion_wins_loses")
stanley_cup_table <- stanley_cup_table[-c(1, 36:98), ]
##trustees
stanley_cup_trustee <-
read_html("https://en.wikipedia.org/wiki/Stanley_Cup")
nodes_stanley_trustee <- html_nodes(stanley_cup_trustee, "table")
stanley_trustee_table <- html_table(nodes_stanley_trustee)[[2]]
colnames(stanley_trustee_table) <- c("trustee_name",
"appointed_year", "served_until", "succeeded")
stanley_trustee_table <- stanley_trustee_table[-1, ]
stanley_cup_table$trustee_1 <-
c(as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
```

```
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]),
as.character(stanley_trustee_table[7, 1]))
stanley_cup_table$trustee_2 <-
c(as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[9, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
```

```
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]),
as.character(stanley_trustee_table[8, 1]))
stanley_cup_table <- stanley_cup_table[-1, ]
stanley_cup_table$champion_wins_loses <- gsub("-", "--",
stanley_cup_table$champion_wins_loses)
write.csv(stanley_cup_table, "G:/My Drive/0. study
abroad/academic/10. 2023 Fall/3. SI 564 SQL &
Databases/Homework/Final Project/0. Datasets/tables/stanley cup/0.
stanley cup.csv")


# for team table
hierarchy <- fromJSON("G:/My Drive/0. study abroad/academic/10. 2023
Fall/6. SurvMeth 727 Fundamentals of Computing and Data Display/2.
Assignments/Final Project/hierarchy.json")
## Atlantics
hierarchy_df_atlantics <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[1]]$tea
ms[[1]])
hierarchy_df_atlantics<- hierarchy_df_atlantics[, -c(7:16)]
hierarchy_df_atlantics[, 7] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_atlantics)[7] <- "id_2"
hierarchy_df_atlantics_venue <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[1]]$tea
ms[[1]][["venue"]])
hierarchy_df_atlantics_venue[, 11] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_atlantics_venue)[11] <- "id_2"
hierarchy_df_atlantics_all <- merge(hierarchy_df_atlantics,
hierarchy_df_atlantics_venue, by="id_2")
hierarchy_df_atlantics_all[, 18] <- c("Atlantic", "Atlantic",
"Atlantic", "Atlantic", "Atlantic", "Atlantic", "Atlantic",
"Atlantic")
hierarchy_df_atlantics_all[, 19] <- c("Eastern", "Eastern",
"Eastern", "Eastern", "Eastern", "Eastern", "Eastern", "Eastern")
colnames(hierarchy_df_atlantics_all)[c(18, 19)] <- c("division",
"conference")
## Metros
```

```
hierarchy_df_metros <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[1]]$tea
ms[[2]])
hierarchy_df_metros <- hierarchy_df_metros[, -c(7:16)]
hierarchy_df_metros[, 7] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_metros)[7] <- "id_2"
hierarchy_df_metros_venue <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[1]]$tea
ms[[2]][["venue"]])
hierarchy_df_metros_venue[, 11] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_metros_venue)[11] <- "id_2"
hierarchy_df_metros_all <- merge(hierarchy_df_metros,
hierarchy_df_metros_venue, by="id_2")
hierarchy_df_metros_all[, 18] <- c("Metropolitan", "Metropolitan",
"Metropolitan", "Metropolitan", "Metropolitan", "Metropolitan",
"Metropolitan", "Metropolitan")
hierarchy_df_metros_all[, 19] <- c("Eastern", "Eastern", "Eastern",
"Eastern", "Eastern", "Eastern", "Eastern", "Eastern")
colnames(hierarchy_df_metros_all)[c(18, 19)] <- c("division",
"conference")
## Pacific
hierarchy_df_pacific <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[2]]$tea
ms[[1]])
hierarchy_df_pacific <- hierarchy_df_pacific[, -c(7:16)]
hierarchy_df_pacific[, 7] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_pacific)[7] <- "id_2"
hierarchy_df_pacific_venue <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[2]]$tea
ms[[1]][["venue"]])
hierarchy_df_pacific_venue[, 11] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_pacific_venue)[11] <- "id_2"
hierarchy_df_pacific_all <- merge(hierarchy_df_pacific,
hierarchy_df_pacific_venue, by="id_2")
hierarchy_df_pacific_all[, 18] <- c("Pacific", "Pacific", "Pacific",
"Pacific", "Pacific", "Pacific", "Pacific", "Pacific")
hierarchy_df_pacific_all[, 19] <- c("Western", "Western", "Western",
"Western", "Western", "Western", "Western", "Western")
colnames(hierarchy_df_pacific_all)[c(18, 19)] <- c("division",
"conference")
## Central
hierarchy_df_central <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[2]]$tea
ms[[2]])
hierarchy_df_central <- hierarchy_df_central[, -c(7:16)]
```

```
hierarchy_df_central[, 7] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_central)[7] <- "id_2"
hierarchy_df_central_venue <-
as.data.frame(hierarchy["conferences"]$conferences$divisions[[2]]$tea
ms[[2]][["venue"]])
hierarchy_df_central_venue[, 11] <- c(1, 2, 3, 4, 5, 6, 7, 8)
colnames(hierarchy_df_central_venue)[11] <- "id_2"
hierarchy_df_central_all <- merge(hierarchy_df_central,
hierarchy_df_central_venue, by="id_2")
hierarchy_df_central_all[, 18] <- c("Central", "Central", "Central",
"Central", "Central", "Central", "Central", "Central")
hierarchy_df_central_all[, 19] <- c("Western", "Western", "Western",
"Western", "Western", "Western", "Western", "Western")
colnames(hierarchy_df_central_all)[c(18, 19)] <- c("division",
"conference")
### merge them all
all_teams <- rbind(hierarchy_df_atlantics_all,
hierarchy_df_metros_all, hierarchy_df_pacific_all,
hierarchy_df_central_all)
all_teams <- all_teams[, -c(1, 2, 6, 7, 8, 17)]
colnames(all_teams) <- c("nickname", "market", "acronym",
"home_arena", "capacity", "address", "city", "state", "zip",
"country", "timezone", "division", "conference")
### export the file to csv
write.csv(all_teams, "G:/My Drive/0. study abroad/academic/10. 2023
Fall/3. SI 564 SQL & Databases/Homework/Final Project/0.
Datasets/tables/team/team_info.csv")
```

C. Presentation Slides

Tables and Data Sources

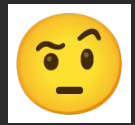
- **player-related**
  - player (Kaggle NHL Database)
  - player_stat (scraped from web)
  - salary (same web as for player_stat)
- **team-related**
  - team (APIs from Sportradar)
  - division (Wikipedia)
  - coach (Wikipedia for current & NHL web for history)
  - arena (same APIs as for team)
- **game-related**
  - game (APIs from Sportradar)
  - stanley_cup (web & wikipedia for trustees)

Data Complexity & Motivators

- 9 data sources contain information at different levels of detail
- Numerous European players are thus excluded from further analysis
- Ice Hockey is fun!!
- ERD
  rapid overview of table connection:

What are the anticipated salaries for each position and handedness?

```
mysql> select avg(s.salary) as average_salary, p.position, p.handedness from salary s join player p on s.player_id = p.player_id where position
 is not null and handedness is not null group by p.position, p.handedness order by average_salary desc;
+----------------+----------+------------+
| average_salary | position | handedness |
+----------------+----------+------------+
|   3007826.0516 | LW       | R          |
|   2692803.7143 | RW       | L          |
|   2289886.6395 | C        | L          |
|   2284128.3771 | D        | R          |
|   2060940.1087 | C        | R          |
|   2045984.3969 | G        | L          |
|   2034368.5238 | D        | L          |
|   2031911.4299 | RW       | R          |
|   1961625.1140 | LW       | L          |
|   1398271.2766 | G        | R          |
+----------------+----------+------------+
10 rows in set (0.11 sec)
```



Envisioning Future Work (this slide will be incorporated to slide 3 during my presentation)

01

Manually encode the IDs of players with incompatible characters from their names to ensure comprehensive inclusion.

02

Spend more time thinking about normalization of the team table to incorporate historical changes in teams in the database.