

### *teamsDB:*

**team\_id:** [taken from NHL api]  
**team\_name:** [NHL team name]  
**status\_id:** [0 = not qualified;  
 1 = eliminated – 1<sup>st</sup> rd  
 2 = eliminated – 2<sup>nd</sup> rd  
 3 = eliminated – 3<sup>rd</sup> rd  
 4 = eliminated – 4<sup>th</sup> rd  
 5 = active]

### *pool\_entry table;*

**PoolTeamName:** [QuadSquad]  
**GM:** [GM Name]  
**Email:**  
**Hometown:**  
**Country:**  
**Pay Status:** [NotPaid, PayPal, Cheque, Cash, Coupon]  
**Pay Amount:**  
**Status:** [Incomplete, complete]

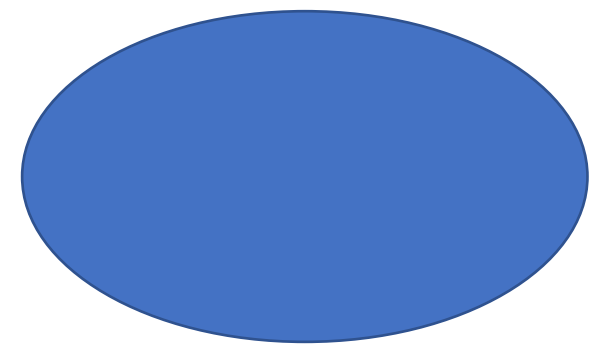
pool\_entry

pool\_team

...

25\*

ID, Name, Team, G, A, W, SO, Pts, State



## Files from July 2020:

HP\_Design.pptx - Design doc

jhp.py – Database modification functions

players-update.py - modifies playersDB (two tables: active\_players, all\_players - also gets data from teamsDB

pool-entry.py – routines for Pool table registration and player selection

teams-update.py - initiates teamsDB and teams table. Records which team is active, etc.

PoolDB

Pool has table

Each poolEntry has a table.

---

## Files as of August 4, 2020.

main.py – calls optimized\_update(). (also calls create\_active\_players\_table(), update\_all\_players(), update\_active\_players())

jhp.py – db/table modification functions: db\_create(), db\_connect(), db\_create\_table(), db\_drop\_table(), table\_exists(), table\_empty()

players.py - modifies players tables (active\_players, all\_players), also gets data from teamDB

- create\_all\_players\_table(), create\_active\_players\_table(), update\_all\_players(), update\_active\_players(), optimized\_update(), get\_player\_stats(), update\_goalie\_stats()

pool.py – creates/inserts pool teams into pool database. - create\_entry\_table(), create\_stats\_table(), create\_roster\_table(), create\_pool\_entry(), add\_player(), update\_pool\_entry(), update\_team\_stats(), update\_all\_team\_stats()

teams.py – Keeps track of active hockey teams and stores status in teams database. Update\_team\_status(), insert\_teams()

test.py - testcreating pool entry.

## Notes:

- The stats only get updated a few hours after the game, supposedly when the data is final. Therefore, the initial version of our DB will not present live data since we are pulling from the api-stats and not the live feed.
- Later on we can parse the live feed and put the stats into a temporary playersDB and allow the pool to draw from this.
- Brainstorming ideas for live feed:
  - Generally, it would be better if we write JHP such that it encourages users to look at the pool while games are active. This will be we encouraged if the following characteristics are programmed:
    - All data is live. (we need to parse the live feed)
    - We need to understand what games will be played today and what games are active.
    - The client could have a personal view (Jakes-Flakes) (or not) and part of the screen (or a second screen) can show what players are actively playing games or playing games later today.
    - Also it would be nice if JHP could tell individual GMs (Jakes\_flakes), how your team would benefit if certain players scored.
    - It would be nice if the pool could produce a personal crib sheet. “today, it would benefit Jakes Flakes is Crosby scored and Thornton didn’t.”
    -