

Getting Started with SDQL

Sports Data Query Language (SDQL) is the code used to query our sports databases. It is for sports enthusiasts, fantasy league players, sports media, and both professional and amateur handicappers. It is similar to SQL programming language but with our unique sports data fields and shortcuts. It allows anyone to investigate past sports results.

There are thousands of types of queries you can build, but here we want to start for the basics.

The format to run a basic query is simple. A query consists of any number of parameters separated by the word “and” or “or.” For simple queries, the order of parameters typically does not matter

Parameters can be thought of as the most common statistics and data used in each sport. For instances *points* is a popular parameter in football and basketball, but in the MLB you would use *runs* and the NHL you would use *goals*. *hits* is a parameter specific to the MLB while *field goals attempted* is a parameter in both the football and basketball databases, but refers to FAR different things in each sport. A list of the available parameters for each sport is available on each databases' [Query Page - HERE](#).

Now you will typically want to attach a numerical value to each parameter in a query. You can do this by using these common mathematical symbols:

= equal to

> greater than

>= greater than or equal to

< less than

<= less than or equal to

!= not equal to

+ addition

- subtraction

* multiply

/ divide

() group together certain parameters and/or change the order of operations

For example, the query for games where a team ran for at least 150 yards is *rushing yards>=150*.

Keep in mind that you may want to use negative values as well. *margin<-24* refers to a team that lost by more than 24 points in this game.

*Note: Most parameters use a numerical input even if you typically do not think of them as numeric based. Typically this will use the coding language of 0=No and 1=Yes. As an example playoffs=0 will mean non-playoff games and playoffs=1 will refer to playoff games. Certain parameters may also have None as the null value. There are exceptions to this rule such as site, where the values may be home, away or neutral. **If you want to know the potential values of a field just run a query on the field by itself (EX: “playoffs” - ENTER)*

Queries can be run looking at an individual team (or player) or all teams in the sport. To query games where the Cleveland Browns scored exactly 30 points last game, use *team=Browns and*

points=30. An NFL query of just *points=30* would show the results for ALL NFL teams in games where they score exactly 30 points.

Whether you are querying a certain team or league-wide results, it is between the team and the opponent. The "team" is the team for which you are showing the parameter results and the "opponent" is the team they are playing that game. To distinguish between the two, use the following prefixes:

t: team (the *t* is optional in most cases)

o: opponent

Going back to that last example, *team=Browns and t:points=30* or *team=Browns and points=30* will show the results in games where the Browns scored 30 points. On the other hand, *team=Browns and o:points=30* will show the results for games where the Browns allowed 30 points. You can also easily compare teams statistics. *rushing yards-o:rushing yards<=-100* pulls up games where the team was outrushed by at least 100 yards.

You can stack together as many parameters as you want in a query. For instances, the query *points>=30 and passing yards>=300 and o:points<7 and o:turnovers=2* will show games where a team scored at least 30 points, passed for at least 300 yards, allowed less than 7 points and forced at least 2 turnovers. The ability to add as many parameters as you wish and fully customize the result for each is part of what makes our Killersports.com databases so powerful!

Much more importantly however is that the SQL does not limit you to searching using data from just the current game. You can search using data from a team's last game, any number of previous games from a team, the opponent's last game or last several games, the last matchup of last several matchups between the two teams or even future games. The following prefixes are what you need to know:

p team's previous game

op opponent's previous game

P previous match up

n team's next game

on opponents' next game

N => next match up

To query results for when a team scored at least 38 points in their last game, use *p:points>=38*. To query results for when a team scored at least 38 points in their last meeting vs this opponent, use *P:points>=38*.

Every *p* you include with the prefix will look back a game further and the same holds true for the *P*, *n* and *N*. This means that *ppp:points<14* refers to a team scoring less than 14 points three games ago and *PPo:rushing yards>=200* refers to a team that allowed at least 200 rushing yards in the second last meeting between these two teams.

There is no limit to how you want to combine these parameters. The query *pp:margin>=10 and p:margin<-17 and Po:points>35* searches for teams that won by at least 10 points two games ago,

lost by more than 17 points last game and allowed more than 35 points the last time they faced the current opponent.

Now you are ready to get started running basic queries! It is important to understand exactly what you have queried and keep in mind it is easy for SQL experts to make mistakes with queries from time-to-time!

Good luck querying!