Solr Query Language Cheatsheet

A quick reference guide to the Solr Query Language (SQL), covering syntax, operators, functions, and best practices for effective searching and data retrieval within Solr.

Basic Syntax & Query Parameters

Core Query Syntax

`q` : Main query parameter. Specifies the search query.
Example: `q=name:product` (finds documents where the 'name' field contains 'product')

`fq` : Filter Query. Restricts the set of documents that can be returned, without influencing the score.
Example: `fq=category:electronics` (filters results to only include documents where 'category' is 'electronics')

`sort` : Sorts the results by a specified field.
Example: `sort=price asc` (sorts results by 'price' in ascending order)

`start` : Specifies the offset of the first result to return.
Example: `start=10` (starts the results from the 11th document)

`rows` : Specifies the number of results to return.
Example: `rows=20` (returns 20 documents per page)

`fl` : Specifies the fields to return in the result.
Example: `fl=id,name,price` (returns only the 'id', 'name', and 'price' fields)

Query Operators

`AND` , `OR` , `NOT`	Boolean operators for combining query terms. **Example:** `q=category:electronics AND brand:Samsung`
`+` (required)	Specifies that a term must be present in the document. **Example:** `q=+name:product +price:[10 TO 100]`
`-` (prohibit)	Excludes documents containing the term. **Example:** `q=category:electronics -brand:Apple`
`:`	Specifies a field for the term. **Example:** `q=name:product`
`*`	Wildcard for matching any characters. **Example:** `q=name:prod*`
`?`	Wildcard for matching a single character. **Example:** `q=name:p?od`

Advanced Querying

Fuzzy Search

Fuzzy searches find terms that are similar to a specified term.
Syntax: `term~distance`
Example: `q=name:roam~2` (finds terms within a Levenshtein distance of 2 from 'roam')

Note: The distance value is optional. If not specified, the default value is 2.

Proximity Search

Proximity searches find terms that are within a specific distance from each other.
Syntax: `"term1 term2"~distance`
Example: `q=name:"quick brown"~5` (finds 'quick' and 'brown' within 5 words of each other in the 'name' field)

Range Queries

Range queries match documents whose field(s) values are between the upper and lower bounds specified.
Syntax: `field:[lower TO upper]`
Example: `q=price:[10 TO 100]` (finds documents where 'price' is between 10 and 100, inclusive)

To exclude the upper or lower bound, use curly brackets `{}` .
Example: `q=price:{10 TO 100}` (finds documents where 'price' is greater than 10 and less than 100)

Boosting

`^` (Boost Factor)	Assigns a higher relevance score to terms, increasing their ranking in the results. **Example:** `q=name:product^2 OR description:product` (boosts documents where 'product' is in the 'name' field)
Note:	The default boost factor is 1. Values greater than 1 increase the score, while values less than 1 decrease the score.

Function Queries & Local Parameters

Function Queries

Function queries allow you to use functions to calculate a score based on field values.
Syntax: `_val_:` followed by the function.
Example: `q={!func}sqrt(popularity)` (scores documents based on the square root of the 'popularity' field)

Common Functions: `sqrt()` , `abs()` , `if()` , `sum()` , `product()`

Local Parameters

Local parameters allow you to modify the behavior of specific query parts.
Syntax: `{!paramName paramValue}`
Example: `q={!type=lucene df=description}product` (searches for 'product' in the 'description' field using the Lucene query parser)

Common Parameters: `type` (query parser), `df` (default field), `v` (value)

Spatial Queries

Solr supports spatial queries to find documents within a certain distance of a point.
Common Parameters: `pt` (point), `d` (distance), `sfield` (spatial field)
Example: `q={!geofilt pt=45.15,-93.85 sfield=location d=5}` (finds documents within 5 kilometers of the point 45.15,-93.85 using the 'location' field)

Solr SQL Interface

Basic SQL Syntax

Solr provides an SQL interface for querying data using standard SQL syntax.
Example: `SELECT id, name, price FROM collection WHERE category = 'electronics' AND price > 50`

SQL Functions

`COUNT(*)`	Counts the total number of documents in the collection. **Example:** `SELECT COUNT(*) FROM collection`
`AVG(field)`	Calculates the average value of a numeric field. **Example:** `SELECT AVG(price) FROM collection`
`SUM(field)`	Calculates the sum of a numeric field. **Example:** `SELECT SUM(price) FROM collection`
`MIN(field)`	Finds the minimum value of a numeric field. **Example:** `SELECT MIN(price) FROM collection`
`MAX(field)`	Finds the maximum value of a numeric field. **Example:** `SELECT MAX(price) FROM collection`

Joins

Solr supports JOIN operations to combine data from multiple collections.

Syntax: `JOIN collection2 ON collection1.field = collection2.field`

Example: `SELECT c1.id, c1.name, c2.category FROM collection1 c1 JOIN collection2 c2 ON c1.category_id = c2.id`