

# Commands Cheat Sheet

## Main commands

<b>dbt init*</b>	Initializes a new dbt project
<b>dbt build</b>	Builds and tests all selected resources (models, seeds, snapshots, tests)
<b>dbt clean*</b>	Deletes artifacts present in the dbt project
<b>dbt compile</b>	Compiles (but does not run) the models in a project
<b>dbt debug*</b>	Debugs dbt connections and projects
<b>dbt deps</b>	Downloads dependencies for a project
<b>dbt docs</b>	Generates documentation for a Project
<b>dbt list*</b>	Lists resources defined in a dbt project
<b>dbt parse*</b>	Parses a project and writes detailed timing info
<b>dbt run</b>	Runs the models in a project
<b>dbt seed</b>	Loads CSV files into the database
<b>dbt snapshot</b>	Executes "snapshot" jobs defined in a project
<b>dbt source</b>	Provides tools for working with source data (including validating that sources are "fresh")
<b>dbt test</b>	Executes tests defined in a project
<b>dbt rpc*</b>	Runs an RPC server that clients can submit queries to
<b>dbt run-operation</b>	Runs arbitrary maintenance SQL against the database

\*CLI Only

## Global CLI Flags

<b>--version*</b>	Returns information about the currently installed version of dbt.
<b>--record-timing-info*</b>	Saves performance profiling information to a file

\*CLI Only

## Other flags

<b>--help</b>	Shows available commands and arguments
<b>--vars</b>	Supplies variables to the project
<b>--full-refresh</b>	Drops existing tables and recreates them
<b>--fail-fast</b>	Enforces a dbt run/test to end up on first error/test failure
<b>--threads</b>	Specifies the number of threads to use while executing models

## dbt docs

<b>generate</b>	Generates your projects's documentation website. Use the <b>--no-compile</b> argument to skip re-compilation
<b>serve</b>	Starts a webserver on port 8000 to serve your documentation locally

## Node selection

### Syntax overview

command	arguments
<b>run</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b> , <b>--defer</b>
<b>test</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b> , <b>--defer</b>
<b>seed</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b>
<b>snapshot</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b>
<b>list</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b> , <b>--resource-type</b>
<b>compile</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b>
<b>freshness</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b>
<b>build</b>	<b>--select (-s)</b> , <b>--exclude</b> , <b>--selector</b> , <b>--resource-type</b> , <b>--defer</b>

In earlier versions, the flag **--models (-m)** was used instead of **--select**. Those commands still support the **--models** flag for backwards compatibility.

### Specifying resources

The **--select** flag accepts one or more arguments. Each argument can be one of compatibility.

- A packages name
- A model name
- A fully-qualified path to a directory of models
- A selection method (**path**, **tag**, **config**, **test\_type**, **test\_name**)

Examples:

- \$ dbt run --select my\_dbt\_project\_name** - all models in your project
- \$ dbt run --select my\_dbt\_model** - a specific model
- \$ dbt run --select path.to.my.models** - all models in a specific directory
- \$ dbt run --select tag:nightly** - models with the "nightly" tag
- \$ dbt run --select path/to/models** - models contained in path/to/models
- \$ dbt test --select test\_type:generic** - run all generic tests
- \$ dbt test --select test\_type:singular** - run all singular tests
- \$ dbt test --select test\_name:unique** - run all instances of the "unique" test

### Excluding models

dbt provides an **--exclude** flag with the same semantics as **--select**. Models specified with the **--exclude** flag will be removed from the set of models selected with **--select**.

Example:

- \$ dbt run --select my\_package.\*+ --exclude my\_package.a\_big\_model+**

## Graph operators

### Plus operator(+)

- \$ dbt run --select my\_model+** - select my\_model and all children
- \$ dbt run --select +my\_model** - select my\_model and all parents
- \$ dbt run --select +my\_model+** - select my\_model, and all of its parents and children

### N-plus operator

- \$ dbt run --select my\_model+1** - select my\_model and its first-degree children
- \$ dbt run --select 2+my\_model** - select my\_model, its first-degree parents, and its second-degree parents ("grandparents")
- \$ dbt run --select 3+my\_model+4** - select my\_model, its parents up to the 3rd degree, and its children down to the 4th degree

### At operator(@)

- \$ dbt run --models @my\_model** - select my\_model, its children, and the parents of its children

### Star operator(\*)

- \$ dbt run --select finance.base.\*** - run all of the models in models/finance/base

## Methods

tag	source	path
package	config	test_type
test_name	state	exposure
metric	result	source_status

## Set operators

### Unions (space-delineated)

- \$ dbt run --select +snowplow\_sessions +fct\_orders** - run snowplow\_sessions, all ancestors of snowplow\_sessions, fct\_orders, and all ancestors of fct\_orders)

### Intersections (comma-separated) - must meet all conditions

- \$ dbt run --select +snowplow\_sessions+fct\_orders** - run all the common ancestors of snowplow\_sessions and fct\_orders
- \$ dbt run --select marts.finance.tag:nightly** - run models that are in the marts/finance subdirectory and tagged nightly, simultaneously
- \$ dbt test --select +fct\_orders,test\_type:generic** - run generic tests of fct\_orders and its ancestors