**Selection methods**

Selection methods are one type of argument for node selection. These methods allow dbt to run commands, such as “dbt run” **specifying methods sharing a common characteristic**. When you use a selection method, **you must insert a colon (:)**after the method name/key and the method value, with no spaces. The possible selection methods are shown in the cheat sheet.

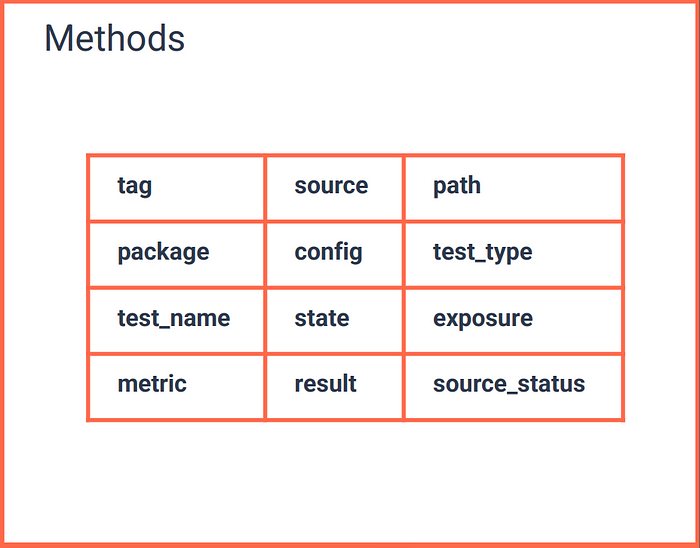


Figure 9 — Methods

Let’s take a look at each method’s description, I will **assume, again, we are using the “dbt run”**command. And since the cheat sheet doesn’t show examples of each method, I will write them here:

* **tag:**Specifying a tag, dbt will run all models **associated with that tag**.  
  - $ dbt run –select tag:monthly
* **source:**Using the source method, dbt will run **all models that select from that source**. It has to be used with the plus (+) operator.  
  - $ dbt run –select source:orders+
* **path and package:**The path method can be used to specify a path, and the package method a package. **It is not mandatory to use these methods**, as seen in the ‘Specifying resources with “ — selection”’ section. You can use them to make your command more verbose.  
  - $ dbt run –select path:path/to/my/model  
  - $ dbt run –select package:my\_package
* **config:**Specifying a config, dbt will run **all models having that config**. The config key is separated from “config” using a dot (.), and the colon separates the config key and value.  
  - $ dbt run –select config.materialized:incremental
* **test\_type:**Test type is used with the “dbt test” command. **It can have one of two arguments**, singular or generic, that specify the type of the test will want to execute.  
  - $ dbt run –select test\_type:generic  
  - $ dbt run –select test\_type:singular
* **test\_name:**Test type is used with the “dbt test” command. This method allows you to **execute all tests with a specific generic name**, such as unique and not\_null.  
  - $ dbt run –select test\_name:not\_null
* **state:**Allows you to run only new nodes (using “new” value) or modified ones (using “modified” value). **Dbt knows if the node is new by looking at the manifest file**. You also can make subselections on modified nodes (check <https://docs.getdbt.com/reference/node-selection/methods#the-state-method>)  
  - $ dbt run –select state:new  
  - $ dbt run –select state:modified
* **exposure and metric:**Specifying an exposure or a metric, dbt will run the **parent's resources of that exposure or metric**. It has to be used with the plus (+) operator left-sided.  
  - $ dbt run –select +exposure:my\_exposure  
  - $ dbt run –select +metric:my\_metric
* **result:**Can be used to **select only resources which resulted in errors** (using “error” value)**or failures** (for tests, using “failure” value) on the prior execution. You can, for example, run only the models which have raised errors on the prior dbt run.  
  - $ dbt run –select result:error
* - $ dbt test –select result:fail
* **source\_status:**Used to execute resources **based on the freshness of the related source**. Check <https://docs.getdbt.com/reference/node-selection/methods#the-source_status-method>.

**Graph operators**

Graph operators are operators which can be used along node selection arguments. For the examples of this section, **consider the DAG shown when I described a node in “Node selection”**.

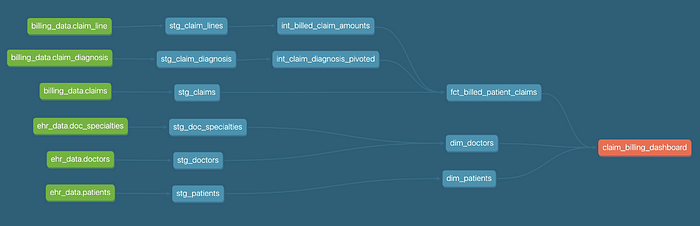


Figure 10 — Example of a dbt’s DAG. Source: <https://github.com/dbt-labs/dbt-project-maturity>

The available graph operators are:

* **Plus operator (+):** Using this operator, the selection **extends to parent resources** (if the plus operator is left-sided) **or to children resources** (if the plus operator is right-sided).  
  - $ dbt run –select +int\_billed\_claim\_amounts

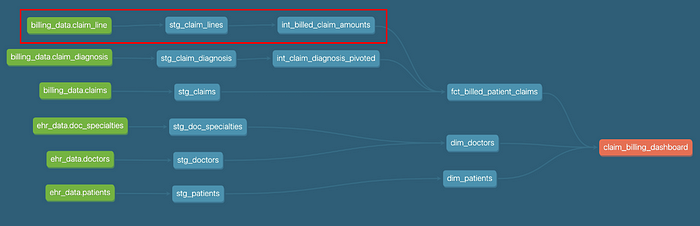


Figure 11 -Current model and all parents are selected

- $ dbt run –select int\_billed\_claim\_amounts+

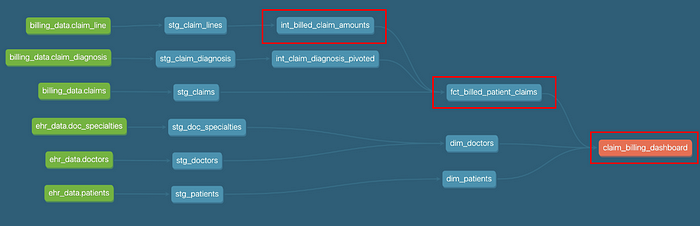


Figure 12 — Current model and all children are selected

- $ dbt run –select +int\_billed\_claim\_amounts+

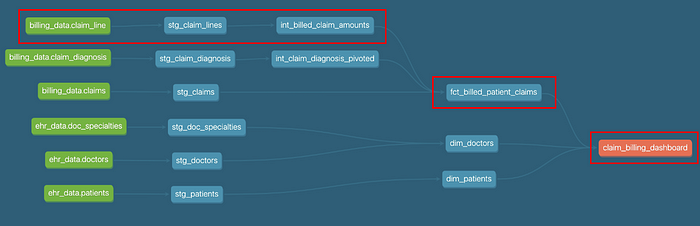


Figure 13 — Current model and all parents and children are selected

* **N-Plus operator:** Similar to plus the operator, but now **you specify the degree** of the parents or children.  
  - $ dbt run –select 1+int\_billed\_claim\_amounts+1

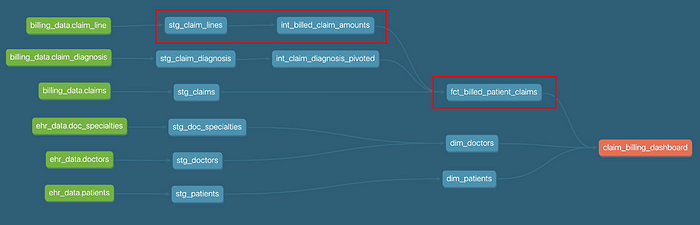
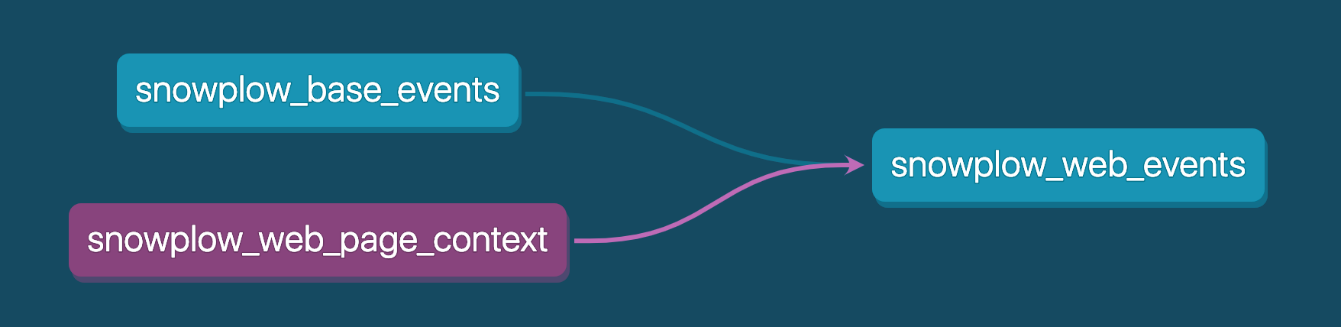


Figure 14 — Current model and parents to the first degree and children to the first degree are selected

* **At operator (@):** Similar to the plus operator, but will **also run the parents of the children of the specified node**.  
  - $ dbt run –select @int\_billed\_claim\_amounts
* The "at" operator[​](https://docs.getdbt.com/reference/node-selection/graph-operators#the-at-operator)

The @ operator is similar to +, but will also include *the parents of the children of the selected model*. This is useful in continuous integration environments where you want to build a model and all of its children, but the *parents* of those children might not exist in the database yet. The selector @snowplow\_web\_page\_context will build all three models shown in the diagram below

[](https://docs.getdbt.com/reference/node-selection/graph-operators)@snowplow\_web\_page\_context will select all of the models shown here

**$ dbt run --models @my\_model**

*# select my\_model, its children, and the parents of its children*

* **Star operator (\*):** Putting this operator at the end of a path, **dbt will execute all nodes in this path**.  
  - $ dbt run –select path.to.models.\*

More examples can be seen in the cheat sheet.

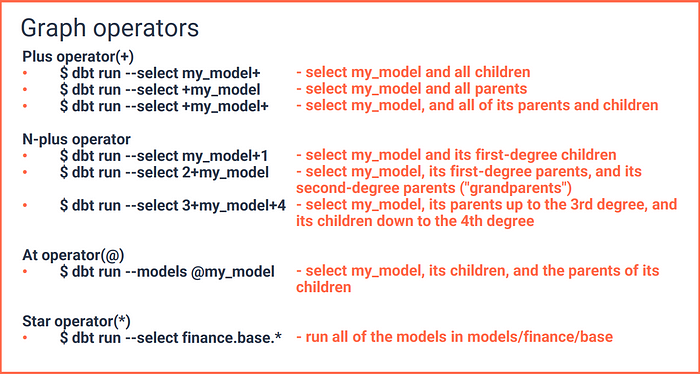


Figure 15 — Graph operators

**Set Operators**

Set operators can be of two types, **unions, and intersections**.

* **Unions:**Unions are used to execute **more than one selection at a time**.

**(Unions are used run multiple models at the same time, they are not linked with eachother or there is no connection between them. They are just supposed to run at the same time)**Using “dbt run” as an example, you can run more than one selection separating them with a blank space.  
- $ dbt run –select +snowplow\_sessions +fact\_orders  
It will run “+snowplow\_session” AND “+fact\_orders”, remember the plus (+) operator runs the parent nodes, so **this command will run both models and their ancestors**.

* **Intersections:**Intersections are used to execute **common resources to more than one selection**. (The intersection is used to run multiple models which have something in common. They may have common parents or they may have common test type: generic /singular or common tags)

The selections are comma-separated (,). Using “dbt run”, for instance, dbt will take the models of each selection and run only the ones that appear in all selections.  
- $ dbt run –select +snowplow\_sessions,+fact\_orders  
It will run the **common ancestors of“snowplow\_session” and “fact\_orders”**.  
- $ dbt run –select marts.finance,tag:nightly  
It will run **models in marts/finance folder AND with the tag nightly, simultaneously.**- $ dbt test — select +fct\_orders,test\_type:generic  
It will run **generic tests of the fct\_orders and its ancestors.**

The **space operator is very useful to run more than one model without needing to divide them into different commands**. And **the comma operator is very convenient if you want to use more than one specifying methods**, or combination of selections.

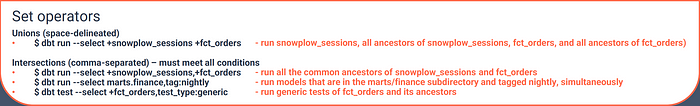


Figure 16 — Set operators

**Excluding models**

**All of the semantics applied to the –selection flag can be applied to the –exclude flag**. Instead of inserting the selected resources in a set that will be executed, the –exclude flag will select resources to remove from this set.

It is a very handy flag, especially if you want to select a large set of resources, but remove some resources that share a common characteristic.

* $ dbt run –select my\_folder.\* –exclude tag:daily  
  It will run all my models inside my\_folder folder, except the ones tagged daily.

