Source Freshness:

A freshness block is used to define the acceptable amount of time between the most recent record, and now, for a [table](https://docs.getdbt.com/terms/table) to be considered "fresh".

In the freshness block, one or both of **warn\_after** and **error\_after** can be provided. If neither is provided, ***then dbt will not calculate freshness snapshots for the tables in this source***.

Additionally, the **loaded\_at\_field**is required to calculate freshness for a table. ***If a loaded\_at\_field is not provided, then dbt will not calculate freshness for the table.***

Freshness blocks are applied hierarchically:

* a freshness and **loaded\_at\_field** property added to a source will be applied to all all tables defined in that source
* a freshness and **loaded\_at\_field** property added to a source *table* will override any properties applied to the source.

*This is useful when all of the tables in a source have the same loaded\_at\_field, as is often the case.*

**Synrax**

**version: 2  
  
sources:  
 - name: <source\_name>  
 freshness:  
 warn\_after:** [**count**](https://docs.getdbt.com/reference/resource-properties/freshness#count)**: <positive\_integer>** [**period**](https://docs.getdbt.com/reference/resource-properties/freshness#period)**: minute | hour | day  
 error\_after:** [**count**](https://docs.getdbt.com/reference/resource-properties/freshness#count)**: <positive\_integer>** [**period**](https://docs.getdbt.com/reference/resource-properties/freshness#period)**: minute | hour | day** [**filter**](https://docs.getdbt.com/reference/resource-properties/freshness#filter)**: <boolean\_sql\_expression>** [**loaded\_at\_field**](https://docs.getdbt.com/reference/resource-properties/freshness#loaded_at_field)**: <column\_name\_or\_expression>  
  
 tables:  
 - name: <table\_name>  
 freshness:  
 warn\_after:** [**count**](https://docs.getdbt.com/reference/resource-properties/freshness#count)**: <positive\_integer>** [**period**](https://docs.getdbt.com/reference/resource-properties/freshness#period)**: minute | hour | day  
 error\_after:** [**count**](https://docs.getdbt.com/reference/resource-properties/freshness#count)**: <positive\_integer>** [**period**](https://docs.getdbt.com/reference/resource-properties/freshness#period)**: minute | hour | day** [**filter**](https://docs.getdbt.com/reference/resource-properties/freshness#filter)**: <boolean\_sql\_expression>** [**loaded\_at\_field**](https://docs.getdbt.com/reference/resource-properties/freshness#loaded_at_field)**: <column\_name\_or\_expression>  
 ...**

loaded\_at\_field

**loaded\_at\_field[​](https://docs.getdbt.com/reference/resource-properties/freshness" \l "loaded_at_field" \o "Direct link to loaded_at_field)** (Required)

It is a column name (or expression) that returns a timestamp indicating freshness.

If using a date field, you may have to cast it to a timestamp:

**loaded\_at\_field:** **"completed\_date::timestamp"**

If using a non-UTC timestamp, cast it to UTC first:

**loaded\_at\_field:** **"convert\_timezone('UTC', 'Australia/Sydney', enter\_column(created\_at\_local))"**

***loaded at field is where we will have to insert the column which consists of the current time\_stamp with date and time. So that the freshness will be check against this and seen if the new data has been inserted so that time frame given in the warn\_after and error\_after. If new data has inserted within the timeframe given in checks, data will be scanned and seen if in that time period new data has been inserted. The freshness check compares the snapshot timestamp of the source data and last time the timestamp of dbt model was run and these are checked and results are taken.***

count[​](https://docs.getdbt.com/reference/resource-properties/freshness#count)

(Required)

A positive integer for the number of periods where a data source is still considered "fresh".

[**count**](https://docs.getdbt.com/reference/resource-properties/freshness#count)**: <positive\_integer>**

period[​](https://docs.getdbt.com/reference/resource-properties/freshness#period)

(Required)

The time period used in the freshness calculation. One of minute, hour or day

[**period**](https://docs.getdbt.com/reference/resource-properties/freshness#period)**: minute | hour | day**

filter[​](https://docs.getdbt.com/reference/resource-properties/freshness#filter)

(optional)

Add a where clause to the query run by dbt source freshness in order to limit data scanned.

[**filter**](https://docs.getdbt.com/reference/resource-properties/freshness#filter)**: <boolean\_sql\_expression>**

This filter *only* applies to dbt's source freshness queries - it will not impact other uses of the source table.

Some databases can have tables where a filter over certain columns are required, in order prevent a full scan of the table, which could be costly. In order to do a freshness check on such tables a filter argument can be added to the configuration, e.g. filter: \_etl\_loaded\_at >= date\_sub(current\_date(), interval 1 day). For the example above, the resulting query would look like

select  
 max(\_etl\_loaded\_at) as max\_loaded\_at,  
 convert\_timezone('UTC', current\_timestamp()) as snapshotted\_at  
from raw.jaffle\_shop.orders  
**where \_etl\_loaded\_at >= date\_sub(current\_date(), interval 1 day)**

This is particularly useful if:

* You are using BigQuery and your source tables are [partitioned tables](https://cloud.google.com/bigquery/docs/partitioned-tables)
* ***You are using Snowflake, Databricks or Spark with large tables, and this results in a performance benefit***

Example:

Version: 2

Sources:

  - name: order\_events

    database: demo

    schema: practice

    tables:

      - name: events\_order

      - name: sales

      - name: sales\_snapshot

        identifier: sales\_src\_freshness

        freshness:

              warn\_after:

                count: 2

                period: minute

              error\_after:

                count: 5

                period: hour

filter: datediff('day', \_etl\_loaded\_at, current\_timestamp) < 2

        loaded\_at\_field: "convert\_timezone('America/Los\_Angeles', 'UTC', created\_time)"

Configuring source freshness output[​](https://docs.getdbt.com/reference/commands/source#configuring-source-freshness-output)

When dbt source freshness completes, a [JSON](https://docs.getdbt.com/terms/json) file containing information about the freshness of your sources will be saved to target/sources.json. An example sources.json will look like:

**target/sources.json**

**{  
 "meta": {  
 "generated\_at": "2019-02-15T00:53:03.971126Z",  
 "elapsed\_time": 0.21452808380126953  
 },  
 "sources": {  
 "source.project\_name.source\_name.table\_name": {  
 "max\_loaded\_at": "2019-02-15T00:45:13.572836+00:00Z",  
 "snapshotted\_at": "2019-02-15T00:53:03.880509+00:00Z",  
 "max\_loaded\_at\_time\_ago\_in\_s": 481.307673,  
 "state": "pass",  
 "criteria": {  
 "warn\_after": {  
 "count": 12,  
 "period": "hour"  
 },  
 "error\_after": {  
 "count": 1,  
 "period": "day"  
 }  
 }  
 }  
 }  
}**

Commands:

1. **Check freshness for all sources and tables:**

dbt source snapshot-freshness

1. **Check freshness for specific sources or tables:**

dbt source snapshot-freshness --select source:my\_source\_name

dbt source snapshot-freshness --select table:my\_source\_name.my\_table\_name

1. **Exclude specific sources or tables from freshness check:**

dbt source snapshot-freshness --exclude source:my\_excluded\_source\_name

dbt source snapshot-freshness –exclude table:my\_excluded\_source\_name.my\_excluded\_table\_name

1. **To override the destination for this sources.json file, use the -o (or --output) flag:**

# Output source freshness info to a different path  
$ **dbt source freshness** **--output target/source\_freshness.json**

Questions

1. What if my source is in a poorly named schema or table?

By default, dbt will use the name: parameters to construct the source reference.

If these names are a little less-than-perfect, use the [schema](https://docs.getdbt.com/reference/resource-properties/schema) and [identifier](https://docs.getdbt.com/reference/resource-properties/identifier) properties to define the names as per the database, and use your name: property for the name that makes sense!

models/<filename>.yml

version: 2  
  
sources:  
 - name: jaffle\_shop  
 schema: postgres\_backend\_public\_schema  
 database: raw  
 tables:  
 - name: orders  
 identifier: api\_orders

In a downstream model:

select \* from {{ source('jaffle\_shop', 'orders') }}

Will get compiled to:

select \* from raw.postgres\_backend\_public\_schema.api\_orders

1. What if my source is in a different database to my target database?

Use the [database property](https://docs.getdbt.com/reference/resource-properties/database) to define the database that the source is in.

models/<filename>.yml

version: 2  
  
sources:  
 - name: jaffle\_shop  
 database: raw  
 tables:  
 - name: orders  
 - name: customers

1. I need to use quotes to select from my source, what should I do?

This is reasonably common on Snowflake in particular.

By default, dbt will not quote the database, schema, or identifier for the source tables that you've specified.

To force dbt to quote one of these values, use the [quoting property](https://docs.getdbt.com/reference/resource-properties/quoting):

models/<filename>.yml

version: 2  
  
sources:  
 - name: jaffle\_shop  
 database: raw  
 quoting:  
 database: true  
 schema: true  
 identifier: true  
  
 tables:  
 - name: order\_items  
 - name: orders  
 *# This overrides the `jaffle\_shop` quoting config*  
 quoting:  
 identifier: false

1. How do I run tests on just my sources?

To run tests on all sources, use the following command:

$ dbt test --select source:\*

(You can also use the -s shorthand here instead of --select)

To run tests on one source (and all of its tables):

$ dbt test --select source:jaffle\_shop

And, to run tests on one source [table](https://docs.getdbt.com/terms/table) only:

$ dbt test --select source:jaffle\_shop.orders

1. How do I run models downstream of one source?

To run models downstream of a source, use the source: selector:

$ dbt run --select source:jaffle\_shop+

(You can also use the -s shorthand here instead of --select)

To run models downstream of one source [table](https://docs.getdbt.com/terms/table):

$ dbt run --select source:jaffle\_shop.orders+

Check out the [model selection syntax](https://docs.getdbt.com/reference/node-selection/syntax) for more examples!

1. How do I exclude a table from a freshness snapshot?

Some tables in a data source may be updated infrequently. If you've set a freshness property at the source level, this [table](https://docs.getdbt.com/terms/table) is likely to fail checks.

To work around this, you can set the table's freshness to null (freshness: null) to "unset" the freshness for a particular table:

models/<filename>.yml

version: 2  
  
sources:  
 - name: jaffle\_shop  
 database: raw  
  
 freshness:  
 warn\_after: {count: 12, period: hour}  
 error\_after: {count: 24, period: hour}  
  
 loaded\_at\_field: \_etl\_loaded\_at  
  
 tables:  
 - name: orders  
 - name: product\_skus  
 freshness: null *# do not check freshness for this table*

***if we want to exclude any table from running into the freshness after defining the source freshness to the sources we will simply add the tables in the same configuration and put freshness: null***

***so that these tables will not be checked for freshness in the table.***

***This is for the reason that in some tables there are chances of data upgradation in frequent time period so checking freshness will anyways call troubles so it can be excluded by this way.***

1. How do I snapshot freshness for one source only?

Use the --select flag to snapshot freshness for specific sources. Eg:

*# Snapshot freshness for all Jaffle Shop tables:*  
$ dbt source freshness --select source:jaffle\_shop  
  
*# Snapshot freshness for a particular source <Term id="table" />:*  
$ dbt source freshness --select source:jaffle\_shop.orders  
  
*# Snapshot freshness for multiple particular source tables:*  
$ dbt source freshness --select source:jaffle\_shop.orders source:jaffle\_shop.customers

See the [source freshness command reference](https://docs.getdbt.com/reference/commands/source) for more information.

1. Are the results of freshness stored anywhere?

Yes!

The dbt source freshness command will output a pass/warning/error status for each [table](https://docs.getdbt.com/terms/table) selected in the freshness snapshot.

Additionally, dbt will write the freshness results to a file in the target/ directory called sources.json by default. You can also override this destination, use the -o flag to the dbt source freshness command.

After enabling source freshness within a job, configure [Artifacts](https://docs.getdbt.com/docs/deploy/artifacts) in your **Project Details** page, which you can find by clicking the gear icon and then selecting **Account settings**. You can see the current status for source freshness by clicking **View Sources** in the job page.