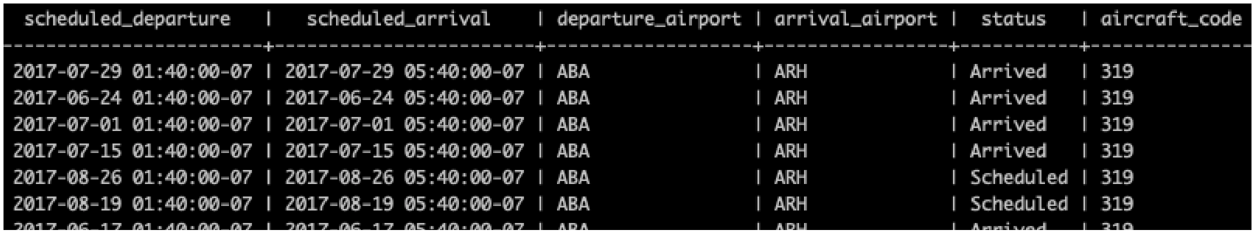
01. Overview

a. I used Postgres Pro's public demo database on Russian airline flight data with 3 months of data from 2017. I have always been very interested in the logistics involved in keeping airlines running on time and balancing having full flights without overbooking. Given there are so many dependencies (getting an aircraft or flight crew to an airport in time for the next flight), it seems like a great optimization problem.

b. For the purposes of this project, I am a data engineer working with a data scientist who is creating a model to forecast air travel demand. The current data is organized at the flight level. However, to conduct her analysis, the data must be restructured at the route level. She also needs some additional variables to feed into her model, such as the distance, occupancy and duration of the route. My hypothesis is structuring the data at the route level will make it easier to identify patterns across flights and reduce the noisiness of the data. A route in this case is all the flights that start and end at the same airport. Only one aircraft type flew each route.

Example: data listed at the flight level, and there are multiple flights for route ABA-ARH



```
 scheduled_departure    |   scheduled_arrival    | departure_airport | arrival_airport |  status   | aircraft_code
------------------------+------------------------+-------------------+-----------------+-----------+--------------
2017-07-29 01:40:00-07 | 2017-07-29 05:40:00-07 | ABA               | ARH             | Arrived   | 319
2017-06-24 01:40:00-07 | 2017-06-24 05:40:00-07 | ABA               | ARH             | Arrived   | 319
2017-07-01 01:40:00-07 | 2017-07-01 05:40:00-07 | ABA               | ARH             | Arrived   | 319
2017-07-15 01:40:00-07 | 2017-07-15 05:40:00-07 | ABA               | ARH             | Arrived   | 319
2017-08-26 01:40:00-07 | 2017-08-26 05:40:00-07 | ABA               | ARH             | Scheduled | 319
2017-08-19 01:40:00-07 | 2017-08-19 05:40:00-07 | ABA               | ARH             | Scheduled | 319
2017-06-17 01:40:00-07 | 2017-06-17 05:40:00-07 | ABA               | ARH             | Arrived   | 319
```

02. Model Structure

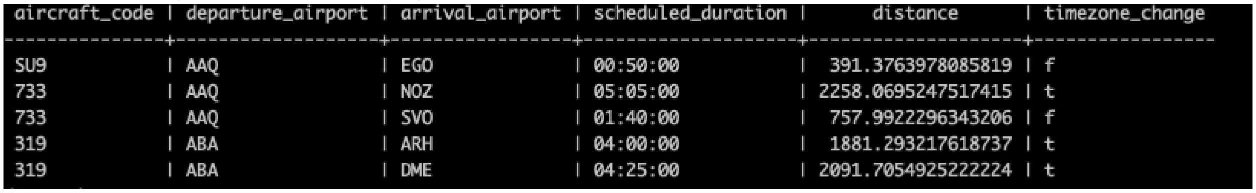
a. Getting my dbt project hooked up to my SQL database took a long time, but writing the data pipeline itself probably took only 3-4 hours. The majority of the time was testing it in SQL and redoing early pieces of the pipeline as I realized what additional data I needed/what data was extraneous. The pipeline takes 14.60 seconds to run. 13.31 seconds of that time was for the incremental view. These measurements were in the dev environment but there were no significant run time differences between dev and prod.

b. Below is a breakdown of each model. The last page has the full table produced by my 5 level deep pipeline. Table snippets added where it felt relevant to understanding the steps taken.

Step1: Allow for some distance based estimates at the route level

Steps:

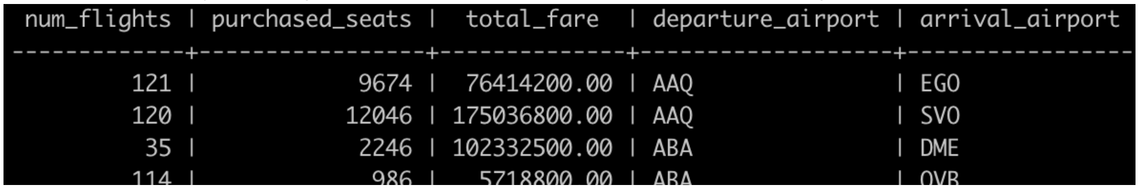
- Rough point distance estimate between airports (not perfect distance of route flown, but approximation)
- Calculate duration from arrival and departure times
- Identify if timezone change occurs via a boolean if arrival timezone = departure timezone



```
aircraft_code | departure_airport | arrival_airport | scheduled_duration |      distance      | timezone_change
--------------+-------------------+-----------------+--------------------+--------------------+-----------------
SU9           | AAQ               | EGO             | 00:50:00           |  391.3763978085819 | f
733           | AAQ               | NOZ             | 05:05:00           | 2258.0695247517415 | t
733           | AAQ               | SVO             | 01:40:00           |  757.9922296343206 | f
319           | ABA               | ARH             | 04:00:00           |  1881.293217618737 | t
319           | ABA               | DME             | 04:25:00           | 2091.7054925222224 | t
```

Step2: Aggregate ticket and flight data at the route level
Steps:

- Count unique flights
- Sum ticket info (seats purchased etc)
- Group by route (aircraft, departure airport, arrival airport)

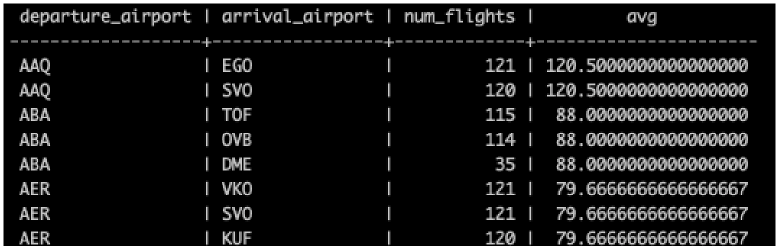


```
 num_flights | purchased_seats |  total_fare   | departure_airport | arrival_airport
-------------+-----------------+---------------+-------------------+-----------------
         121 |            9674 |  76414200.00  | AAQ               | EGO
         120 |           12046 | 175036800.00  | AAQ               | SVO
          35 |            2246 | 102332500.00  | ABA               | DME
         114 |             986 |   5718800.00  | ABA               | OVB
```

Step3: Get available seats to determine occupancy
Steps: Use aircraft info to determine number of seats available per aircraft, multiply by number of flights to get total number of seats.

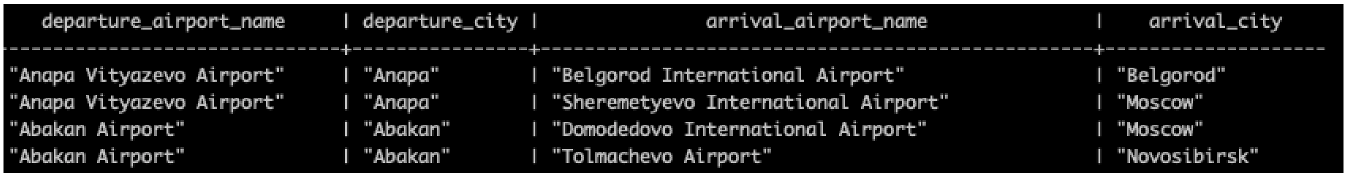
Step4: Calculate occupancy per route
Steps: Add a column calculating occupancy (purchased seats / total seats)

Step5: Determine busiest average departure airports
Steps: Use a window function to partition avg num_flights on departure airport



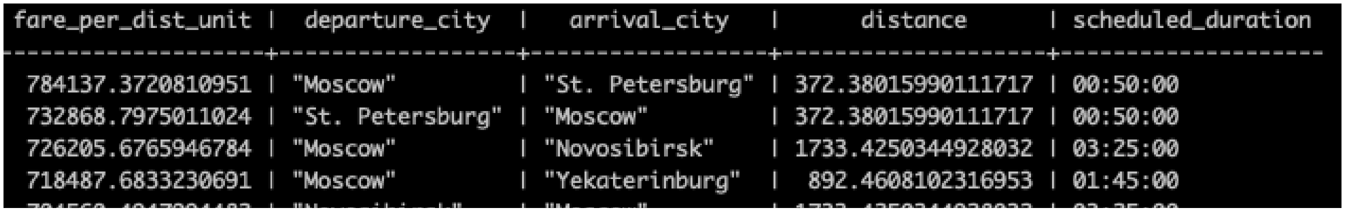
```
 departure_airport | arrival_airport | num_flights |         avg
-------------------+-----------------+-------------+----------------------
 AAQ               | EGO             |         121 | 120.5000000000000000
 AAQ               | SVO             |         120 | 120.5000000000000000
 ABA               | TOF             |         115 |  88.0000000000000000
 ABA               | OVB             |         114 |  88.0000000000000000
 ABA               | DME             |          35 |  88.0000000000000000
 AER               | VKO             |         121 |  79.6666666666666667
 AER               | SVO             |         121 |  79.6666666666666667
 AER               | KUF             |         120 |  79.6666666666666667
```

Step6: Make it easy to interpret data
Steps: Add english name of airport and city, extracted from JSON. For readability this only includes the new columns but these were added to all of the additional columns in step4.



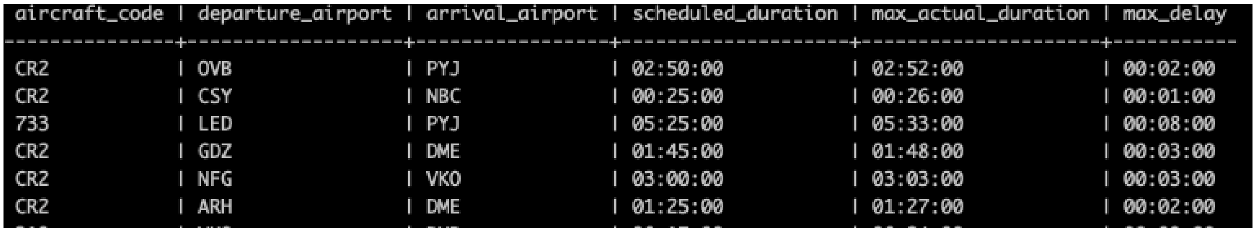
```
   departure_airport_name     | departure_city |               arrival_airport_name               |    arrival_city
------------------------------+----------------+--------------------------------------------------+--------------------
"Anapa Vityazevo Airport"     | "Anapa"        | "Belgorod International Airport"                 | "Belgorod"
"Anapa Vityazevo Airport"     | "Anapa"        | "Sheremetyevo International Airport"             | "Moscow"
"Abakan Airport"              | "Abakan"       | "Domodedovo International Airport"               | "Moscow"
"Abakan Airport"              | "Abakan"       | "Tolmachevo Airport"                             | "Novosibirsk"
```

Lucrative_Routes: Find the routes with the highest fare per distance unit
Steps: Select the top 10 routes with the highest fare per distance unit measurements



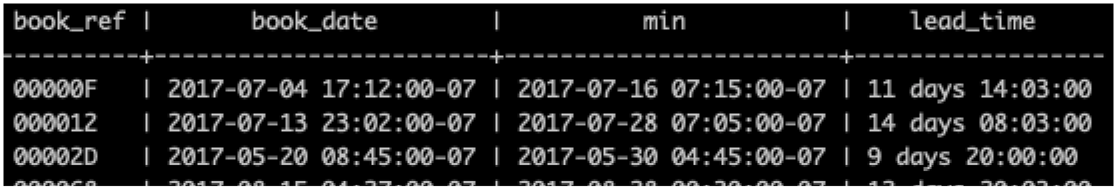
```
 fare_per_dist_unit |  departure_city   |   arrival_city    |      distance      | scheduled_duration
--------------------+-------------------+-------------------+--------------------+--------------------
  784137.3720810951 | "Moscow"          | "St. Petersburg"  | 372.38015990111717 | 00:50:00
  732868.7975011024 | "St. Petersburg"  | "Moscow"          | 372.38015990111717 | 00:50:00
  726205.6765946784 | "Moscow"          | "Novosibirsk"     | 1733.4250344928032 | 03:25:00
  718487.6833230691 | "Moscow"          | "Yekaterinburg"   |  892.4608102316953 | 01:45:00
```

Delays: Find the routes with max delays, among flights with actual data
Steps: Compare scheduled with actual duration, take max per route



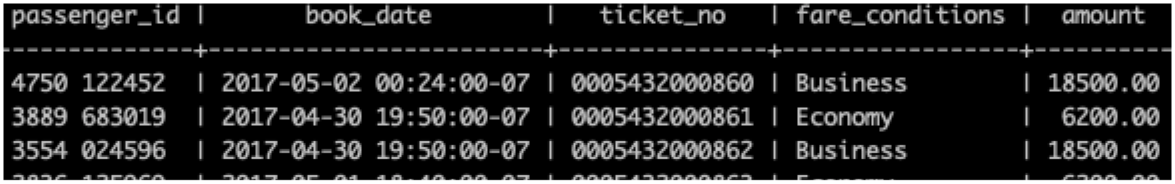
```
 aircraft_code | departure_airport | arrival_airport | scheduled_duration | max_actual_duration | max_delay
---------------+-------------------+-----------------+--------------------+---------------------+-----------
 CR2           | OVB               | PYJ             | 02:50:00           | 02:52:00            | 00:02:00
 CR2           | CSY               | NBC             | 00:25:00           | 00:26:00            | 00:01:00
 733           | LED               | PYJ             | 05:25:00           | 05:33:00            | 00:08:00
 CR2           | GDZ               | DME             | 01:45:00           | 01:48:00            | 00:03:00
 CR2           | NFG               | VKO             | 03:00:00           | 03:03:00            | 00:03:00
 CR2           | ARH               | DME             | 01:25:00           | 01:27:00            | 00:02:00
```

Booking_Leadtime: Create clear picture of how far in advance people purchased tickets
Steps: Compare booking date to min flight departure date, calculate difference



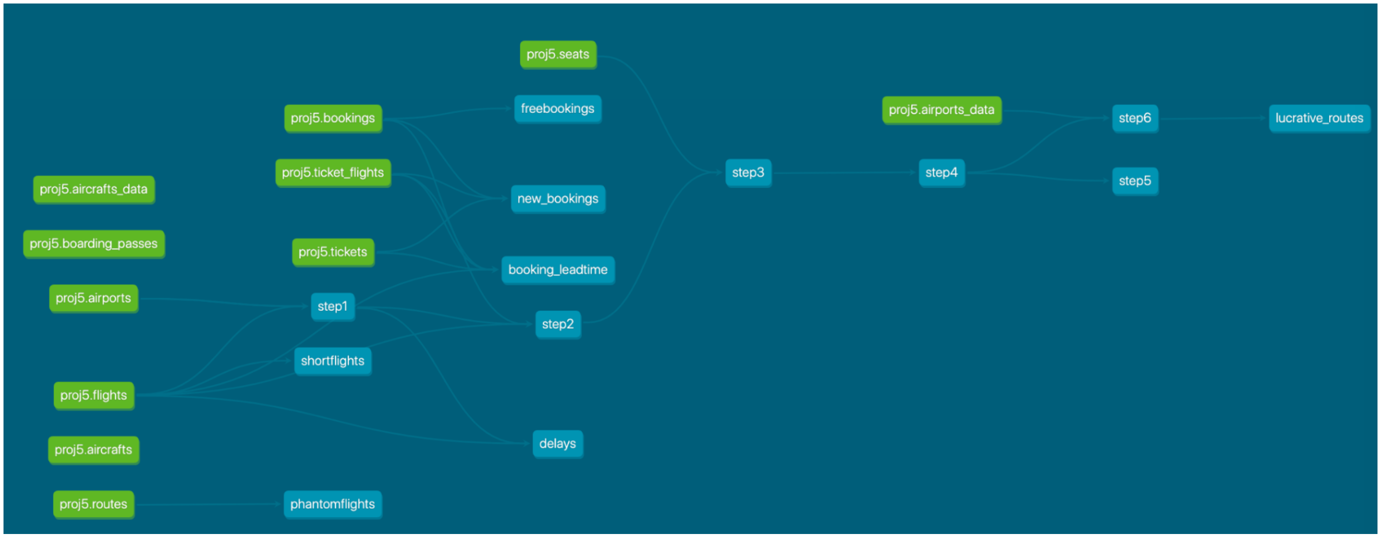
```
 book_ref |       book_date        |          min           |    lead_time
----------+------------------------+------------------------+-----------------
 00000F   | 2017-07-04 17:12:00-07 | 2017-07-16 07:15:00-07 | 11 days 14:03:00
 000012   | 2017-07-13 23:02:00-07 | 2017-07-28 07:05:00-07 | 14 days 08:03:00
 00002D   | 2017-05-20 08:45:00-07 | 2017-05-30 04:45:00-07 | 9 days 20:00:00
```

New_Bookings: Identify new bookings that haven't yet been analyzed
Steps: Create incremental view of most recent bookings



```
 passenger_id |       book_date        |   ticket_no    | fare_conditions |  amount
--------------+------------------------+----------------+-----------------+----------
 4750 122452  | 2017-05-02 00:24:00-07 | 0005432000860  | Business        | 18500.00
 3889 683019  | 2017-04-30 19:50:00-07 | 0005432000861  | Economy         |  6200.00
 3554 024596  | 2017-04-30 19:50:00-07 | 0005432000862  | Business        | 18500.00
```

c. See schemas.yml for descriptions
d. Lineage DAG diagram



e. Luckily for me, the creators of this dataset included a number of constraints that meant the data was pretty clean when it got to me. As such, I tried to focus on situations that might be valid data types but not logical (things a human would immediately find suspicious but a computer might not). I came up with three tests to validate the data behaved how we'd expect of an airline.With more time, I'd like to add some similar tests for newly created fields. For example, occupancy rate should never exceed 1, and booking lead time should never be negative.

- Phantom flights: ensuring no flights took off and landed from the same airport
- Free bookings: no tickets were booked for $0
- Short flights: there were no flights under 10 minutes. This could throw off duration calculations down the line.

03. Interesting Findings (Optional)

a. I was very interested to see how the actual duration differed from the scheduled duration. However, for most flights this data was null. For the data available, the greatest difference was 16 minutes. That struck me as surprisingly low given some of the flights were 8 hours long. I excluded this data from my 5 depth model pipeline given the amount of missing data and this bizarre result, which made me think the data available might be systematically different from the data that was missing.

04. Reflection

a. Originally I envisioned this data being used by a data scientist, but I underestimated how much less data I'd have after rolling it up (70k rows to 450). More likely, this data would be useful to an operations or business analyst trying to get a birds eye view of the airline's opportunities. I also took the data schema at its word, and found it was occasionally aspirational. Data in the schema was missing or incomplete, so that caused me to change some plans (for example, not calculating the median delay because the actual flight data was spotty and the status field contradictory).

b. Now that I have this pipeline in place I'd love to dig into the analysis. How far in advance do people typically buy tickets (bookings_leadtime) and does this vary based on the length of the trip? Additionally, I'd like to do more with the seat class - do people buy business class tickets farther in advance than economy seats? Also looking at occupancy by class would be very cool.

c. I enjoyed DBT a lot. In the past I've used SQL stored procedures to manage basic data manipulations and this was definitely more modular and easier to iterate with. Also the doc serve functionality was delightful. I never worked out a great solution for building in dbt and testing in SQL without rewriting a lot of code, so if I were to use it again I'd probably want a better SQL solution than just running in the command line, but otherwise I would definitely use it again.

d. I would prefer to build and write these models and specs in the command line/Visual Studio rather than a GUI for the flexibility it provides. However, I do see value in a GUI particularly for analyst level users whose priority is exploring the existing data rather than creating new pipelining. And I think there would be value to a low-code solution that retained the flexibility of the command line but brought in more visuals.

05. Link to Code

a. Github https://github.com/kailinkoch/airline-dbt

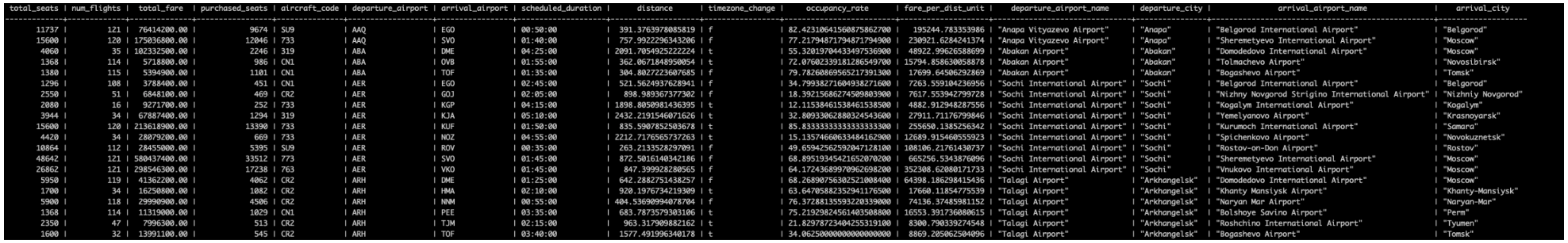
06. Sources

a. Dataset Information: https://postgrespro.com/docs/postgrespro/10/demodb-bookings
b. I used the medium dataset here
c. Other resources

i. https://www.postgresql.org/docs/13/app-psql.html
ii. http://postgresguide.com/utilities/psql.html
iii. https://stackoverflow.com/questions/40865564/why-command-dt-gives-no-relations-found
iv. https://stackoverflow.com/questions/3393961/how-to-import-existing-sql-files-in-postgresql-8-4
v. https://stackoverflow.com/questions/1213430/how-to-fully-delete-a-git-repository-created-with-init

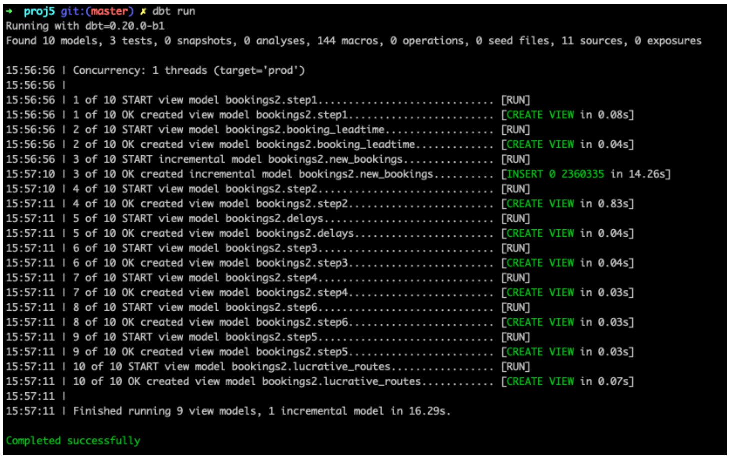
Appendix Images on Next Pages

Final Pipeline Full Table (Rotate and zoom to see clearer)

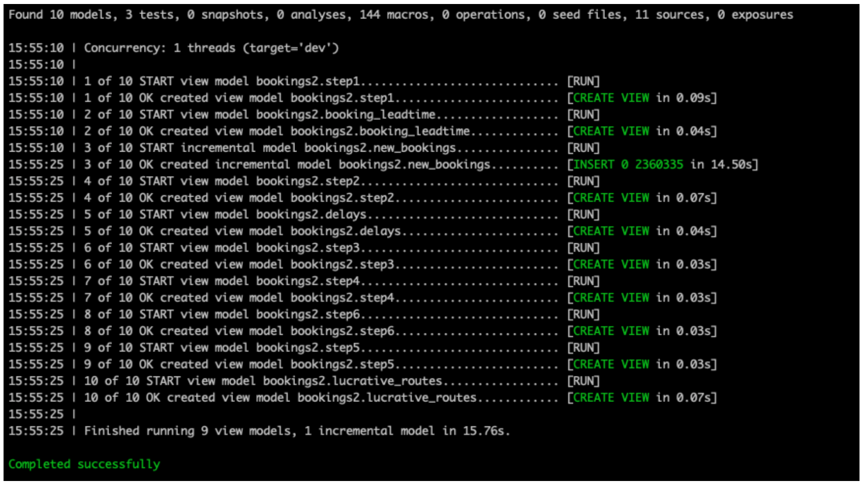


```
 total_seats | num_flights |  total_fare  | purchased_seats | aircraft_code | departure_airport | arrival_airport | scheduled_duration |      distance      | timezone_change |     occupancy_rate      | fare_per_dist_unit |    departure_airport_name     | departure_city |               arrival_airport_name               |    arrival_city
-------------+-------------+--------------+-----------------+---------------+-------------------+-----------------+--------------------+--------------------+-----------------+-------------------------+--------------------+-------------------------------+----------------+--------------------------------------------------+--------------------
       11737 |         121 |  76414200.00 |            9674 | SU9           | AAQ               | EGO             | 00:50:00           |  391.3763978085819 | f               | 82.42310641560875862700 |   195244.783353986 | "Anapa Vityazevo Airport"     | "Anapa"        | "Belgorod International Airport"                 | "Belgorod"
       15600 |         120 | 175036800.00 |           12046 | 733           | AAQ               | SVO             | 01:40:00           |  757.9922296343206 | f               | 77.21794871794871794900 |  230921.6284241374 | "Anapa Vityazevo Airport"     | "Anapa"        | "Sheremetyevo International Airport"             | "Moscow"
        4060 |          35 | 102332500.00 |            2246 | 319           | ABA               | DME             | 04:25:00           | 2091.7054925222224 | t               | 55.32019704433497536900 |  48922.99626588699 | "Abakan Airport"              | "Abakan"       | "Domodedovo International Airport"               | "Moscow"
        1368 |         114 |   5718800.00 |             986 | CN1           | ABA               | OVB             | 01:55:00           |  362.0671848950054 | t               | 72.07602339181286549700 | 15794.858630058878 | "Abakan Airport"              | "Abakan"       | "Tolmachevo Airport"                             | "Novosibirsk"
        1380 |         115 |   5394900.00 |            1101 | CN1           | ABA               | TOF             | 01:35:00           |  304.8027223607685 | f               | 79.78260869565217391300 |  17699.64506292869 | "Abakan Airport"              | "Abakan"       | "Bogashevo Airport"                              | "Tomsk"
        1296 |         108 |   3788400.00 |             451 | CN1           | AER               | EGO             | 02:45:00           |  521.5624937628941 | f               | 34.79938271604938271600 |  7263.559104236956 | "Sochi International Airport" | "Sochi"        | "Belgorod International Airport"                 | "Belgorod"
        2550 |          51 |   6848100.00 |             469 | CR2           | AER               | GOJ             | 02:05:00           |   898.989367377302 | f               | 18.39215686274509803900 |  7617.553942799728 | "Sochi International Airport" | "Sochi"        | "Nizhny Novgorod Strigino International Airport" | "Nizhniy Novgorod"
        2080 |          16 |   9271700.00 |             252 | 733           | AER               | KGP             | 04:15:00           | 1898.8050981436395 | t               | 12.11538461538461538500 |  4882.912948287556 | "Sochi International Airport" | "Sochi"        | "Kogalym International Airport"                  | "Kogalym"
        3944 |          34 |  67887400.00 |            1294 | 319           | AER               | KJA             | 05:10:00           | 2432.2191546071626 | t               | 32.80933062880324543600 |  27911.71176799846 | "Sochi International Airport" | "Sochi"        | "Yemelyanovo Airport"                            | "Krasnoyarsk"
       15600 |         120 | 213618900.00 |           13390 | 733           | AER               | KUF             | 01:50:00           |  835.5907852503678 | t               | 85.83333333333333333300 |  255650.1385256342 | "Sochi International Airport" | "Sochi"        | "Kurumoch International Airport"                 | "Samara"
        4420 |          34 |  28079200.00 |             669 | 733           | AER               | NOZ             | 04:55:00           | 2212.7176565737263 | t               | 15.13574660633484162900 | 12689.915460555923 | "Sochi International Airport" | "Sochi"        | "Spichenkovo Airport"                            | "Novokuznetsk"
       10864 |         112 |  28455000.00 |            5395 | SU9           | AER               | ROV             | 00:35:00           |  263.2133528297091 | f               | 49.65942562592047128100 | 108106.21761430737 | "Sochi International Airport" | "Sochi"        | "Rostov-on-Don Airport"                          | "Rostov"
       48642 |         121 | 580437400.00 |           33512 | 773           | AER               | SVO             | 01:45:00           |  872.5016140342186 | f               | 68.89519345421652070200 |  665256.5343876096 | "Sochi International Airport" | "Sochi"        | "Sheremetyevo International Airport"             | "Moscow"
       26862 |         121 | 298546300.00 |           17238 | 763           | AER               | VKO             | 01:45:00           |   847.399928280565 | f               | 64.17243689970962698200 | 352308.62080171733 | "Sochi International Airport" | "Sochi"        | "Vnukovo International Airport"                  | "Moscow"
        5950 |         119 |  41362200.00 |            4062 | CR2           | ARH               | DME             | 01:25:00           |  642.2882751438257 | f               | 68.26890756302521008400 | 64398.186298415436 | "Talagi Airport"              | "Arkhangelsk"  | "Domodedovo International Airport"               | "Moscow"
        1700 |          34 |  16250800.00 |            1082 | CR2           | ARH               | HMA             | 02:10:00           |  920.1976734219309 | t               | 63.64705882352941176500 |  17660.11854775539 | "Talagi Airport"              | "Arkhangelsk"  | "Khanty Mansiysk Airport"                        | "Khanty-Mansiysk"
        5900 |         118 |  29990900.00 |            4506 | CR2           | ARH               | NNM             | 00:55:00           | 404.53690994078704 | f               | 76.37288135593220339000 |  74136.37485981152 | "Talagi Airport"              | "Arkhangelsk"  | "Naryan Mar Airport"                             | "Naryan-Mar"
        1368 |         114 |  11319000.00 |            1029 | CN1           | ARH               | PEE             | 03:35:00           |  683.7873579303106 | t               | 75.21929824561403508800 | 16553.391736080615 | "Talagi Airport"              | "Arkhangelsk"  | "Bolshoye Savino Airport"                        | "Perm"
        2350 |          47 |   7996300.00 |             513 | CR2           | ARH               | TJM             | 02:15:00           |   963.317909882162 | t               | 21.82978723404255319100 |  8300.790339274548 | "Talagi Airport"              | "Arkhangelsk"  | "Roshchino International Airport"                | "Tyumen"
        1600 |          32 |  13991100.00 |             545 | CR2           | ARH               | TOF             | 03:40:00           |  1577.491996340178 | t               | 34.06250000000000000000 |  8869.205062504096 | "Talagi Airport"              | "Arkhangelsk"  | "Bogashevo Airport"                              | "Tomsk"
```

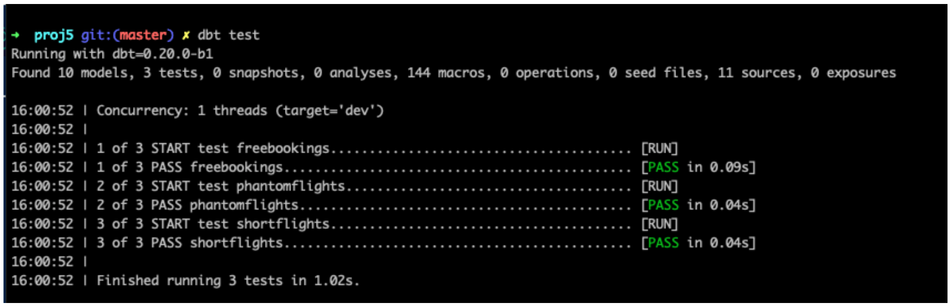
Working in Dev and Prod



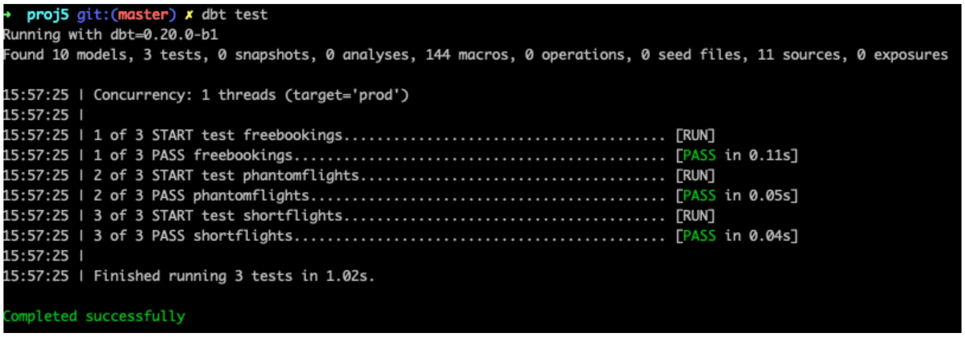
```
➜  proj5 git:(master) ✗ dbt run
Running with dbt=0.20.0-b1
Found 10 models, 3 tests, 0 snapshots, 0 analyses, 144 macros, 0 operations, 0 seed files, 11 sources, 0 exposures

15:56:56 | Concurrency: 1 threads (target='prod')
15:56:56 |
15:56:56 | 1 of 10 START view model bookings2.step1............................ [RUN]
15:56:56 | 1 of 10 OK created view model bookings2.step1....................... [CREATE VIEW in 0.08s]
15:56:56 | 2 of 10 START view model bookings2.booking_leadtime................. [RUN]
15:56:56 | 2 of 10 OK created view model bookings2.booking_leadtime............ [CREATE VIEW in 0.04s]
15:56:56 | 3 of 10 START incremental model bookings2.new_bookings.............. [RUN]
15:57:10 | 3 of 10 OK created incremental model bookings2.new_bookings......... [INSERT 0 2360335 in 14.26s]
15:57:10 | 4 of 10 START view model bookings2.step2............................ [RUN]
15:57:11 | 4 of 10 OK created view model bookings2.step2....................... [CREATE VIEW in 0.83s]
15:57:11 | 5 of 10 START view model bookings2.delays........................... [RUN]
15:57:11 | 5 of 10 OK created view model bookings2.delays...................... [CREATE VIEW in 0.04s]
15:57:11 | 6 of 10 START view model bookings2.step3............................ [RUN]
15:57:11 | 6 of 10 OK created view model bookings2.step3....................... [CREATE VIEW in 0.04s]
15:57:11 | 7 of 10 START view model bookings2.step4............................ [RUN]
15:57:11 | 7 of 10 OK created view model bookings2.step4....................... [CREATE VIEW in 0.03s]
15:57:11 | 8 of 10 START view model bookings2.step6............................ [RUN]
15:57:11 | 8 of 10 OK created view model bookings2.step6....................... [CREATE VIEW in 0.03s]
15:57:11 | 9 of 10 START view model bookings2.step5............................ [RUN]
15:57:11 | 9 of 10 OK created view model bookings2.step5....................... [CREATE VIEW in 0.03s]
15:57:11 | 10 of 10 START view model bookings2.lucrative_routes................ [RUN]
15:57:11 | 10 of 10 OK created view model bookings2.lucrative_routes........... [CREATE VIEW in 0.07s]
15:57:11 |
15:57:11 | Finished running 9 view models, 1 incremental model in 16.29s.

Completed successfully
```



```
Found 10 models, 3 tests, 0 snapshots, 0 analyses, 144 macros, 0 operations, 0 seed files, 11 sources, 0 exposures

15:55:10 | Concurrency: 1 threads (target='dev')
15:55:10 |
15:55:10 | 1 of 10 START view model bookings2.step1............................ [RUN]
15:55:10 | 1 of 10 OK created view model bookings2.step1....................... [CREATE VIEW in 0.09s]
15:55:10 | 2 of 10 START view model bookings2.booking_leadtime................. [RUN]
15:55:10 | 2 of 10 OK created view model bookings2.booking_leadtime............ [CREATE VIEW in 0.04s]
15:55:10 | 3 of 10 START incremental model bookings2.new_bookings.............. [RUN]
15:55:25 | 3 of 10 OK created incremental model bookings2.new_bookings......... [INSERT 0 2360335 in 14.50s]
15:55:25 | 4 of 10 START view model bookings2.step2............................ [RUN]
15:55:25 | 4 of 10 OK created view model bookings2.step2....................... [CREATE VIEW in 0.07s]
15:55:25 | 5 of 10 START view model bookings2.delays........................... [RUN]
15:55:25 | 5 of 10 OK created view model bookings2.delays...................... [CREATE VIEW in 0.04s]
15:55:25 | 6 of 10 START view model bookings2.step3............................ [RUN]
15:55:25 | 6 of 10 OK created view model bookings2.step3....................... [CREATE VIEW in 0.03s]
15:55:25 | 7 of 10 START view model bookings2.step4............................ [RUN]
15:55:25 | 7 of 10 OK created view model bookings2.step4....................... [CREATE VIEW in 0.03s]
15:55:25 | 8 of 10 START view model bookings2.step6............................ [RUN]
15:55:25 | 8 of 10 OK created view model bookings2.step6....................... [CREATE VIEW in 0.03s]
15:55:25 | 9 of 10 START view model bookings2.step5............................ [RUN]
15:55:25 | 9 of 10 OK created view model bookings2.step5....................... [CREATE VIEW in 0.03s]
15:55:25 | 10 of 10 START view model bookings2.lucrative_routes................ [RUN]
15:55:25 | 10 of 10 OK created view model bookings2.lucrative_routes........... [CREATE VIEW in 0.07s]
15:55:25 |
15:55:25 | Finished running 9 view models, 1 incremental model in 15.76s.

Completed successfully
```



```
➜  proj5 git:(master) ✗ dbt test
Running with dbt=0.20.0-b1
Found 10 models, 3 tests, 0 snapshots, 0 analyses, 144 macros, 0 operations, 0 seed files, 11 sources, 0 exposures

16:00:52 | Concurrency: 1 threads (target='dev')
16:00:52 |
16:00:52 | 1 of 3 START test freebookings...................................... [RUN]
16:00:52 | 1 of 3 PASS freebookings............................................ [PASS in 0.09s]
16:00:52 | 2 of 3 START test phantomflights.................................... [RUN]
16:00:52 | 2 of 3 PASS phantomflights.......................................... [PASS in 0.04s]
16:00:52 | 3 of 3 START test shortflights...................................... [RUN]
16:00:52 | 3 of 3 PASS shortflights............................................ [PASS in 0.04s]
16:00:52 |
16:00:52 | Finished running 3 tests in 1.02s.
```



```
➜  proj5 git:(master) ✗ dbt test
Running with dbt=0.20.0-b1
Found 10 models, 3 tests, 0 snapshots, 0 analyses, 144 macros, 0 operations, 0 seed files, 11 sources, 0 exposures

15:57:25 | Concurrency: 1 threads (target='prod')
15:57:25 |
15:57:25 | 1 of 3 START test freebookings...................................... [RUN]
15:57:25 | 1 of 3 PASS freebookings............................................ [PASS in 0.11s]
15:57:25 | 2 of 3 START test phantomflights.................................... [RUN]
15:57:25 | 2 of 3 PASS phantomflights.......................................... [PASS in 0.05s]
15:57:25 | 3 of 3 START test shortflights...................................... [RUN]
15:57:25 | 3 of 3 PASS shortflights............................................ [PASS in 0.04s]
15:57:25 |
15:57:25 | Finished running 3 tests in 1.02s.

Completed successfully
```