

Part 3 – Data Build Tools Demo #2

1. Tugas merupakan lanjutan dari tugas “Data Build Tools Demo #1”, maka dari itu langkah pertama adalah masuk ke file “dbt_project.yml”, lalu buatlah query seperti di bawah ini :

```
my_project > ! dbt_project.yml
32  models:
33    my_project:
34      # Config indicated by + and applies to all files under models/example/
35      _stg:
36        +materialized: view
37        +schema: _stg
38        +database: store
39      _int:
40        +materialized: table
41        +schema: _int
42        +database: store
43      _fct:
44        +materialized: table
45        +schema: _fct
46        +database: store
47      mart:
48        +materialized: table
49        +schema: mart
50        +database: store
```

2. Kemudian buat folder “_fct, dan mart” di dalam folder models.
3. Lalu buat file dengan nama “int_order_details.sql” pada folder “_int” dan masukkan query seperti gambar di bawah ini :

```
my_project > models > _int > int_order_details.sql
1  select
2    details.order_detail_id
3    , details.order_id
4    , orders.order_at
5    , orders.customer_phone
6    , products.brand_id
7    , products.brand_name
8    , details.product_id
9    , products.product_name
10   , products.product_price
11   , details.order_qty
12   , details.unit_sales
13 from {{ ref('stg_order_details') }} as details
14 left join {{ ref('int_orders') }} as orders
15   on details.order_id = orders.order_id
16 left join {{ ref('int_products') }} as products
17   on details.product_id = products.product_id
```

4. Masih di folder “_int”, buatlah file dengan nama “int_orders.sql” dan buat query seperti berikut ini :

```
my_project > models > _int > int_orders.sql
1  select *
2  from {{ ref('stg_orders') }}
```

5. Masih di folder “_int”, buatlah file dengan nama “int_products.sql” dan buat query seperti berikut ini :

```
my_project > models > _int > int_products.sql
1  select
2      products.product_id
3      , brands.brand_id
4      , brands.brand_name
5      , products.product_name
6      , products.product_price
7  from {{ ref('stg_products') }} as products
8  left join {{ ref('stg_brands') }} as brands
9      on products.brand_id = brands.brand_id
```

6. Buat file dengan nama “int_schema.yml”, dan buat query seperti berikut ini :

```
my_project > models > _int > ! int_schema.yml
1  version: 2
2
3  models:
4    - name: int_orders
5      columns:
6        - name: order_id
7          tests:
8            - not_null
9            - unique
10
11    - name: int_products
12      columns:
13        - name: product_name
14          tests:
15            - not_null
16            - unique
17
18        - name: brand_name
19          tests:
20            - not_null
21
22    - name: int_order_details
23      columns:
24        - name: order_detail_id
25          tests:
26            - not_null
27            - unique
28
29      tests:
30        - dbt_expectations.expect_table_row_count_to_equal_other_table:
31            compare_model: ref("stg_order_details")
```

7. Setelah itu jalankan serta uji DBT yang sudah dibuat dengan menggunakan perintah “dbt run && dbt test” pada terminal.

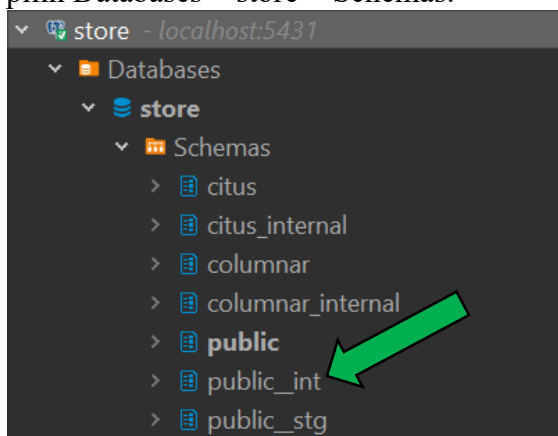
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS bash - my_project + - [ ] [ ] ... [ ] [ ] X

(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
• $ dbt run && dbt test
11:17:51 Running with dbt=1.8.5
11:17:51 Registered adapter: postgres=1.8.2
11:17:52 [WARNING]: Configuration paths exist in your dbt_project.yml file which do not apply to any resources.
There are 2 unused configuration paths:
- models.my_project._fct
- models.my_project.mart
11:17:52 Found 7 models, 33 data tests, 4 sources, 417 macros
11:17:52
11:17:52 Concurrency: 1 threads (target='dev')
11:17:52
11:17:52 1 of 7 START sql table model public__stg.stg_brands ..... [RUN]
11:17:52 1 of 7 OK created sql table model public__stg.stg_brands ..... [SELECT 3 in 0.18s]
11:17:52 2 of 7 START sql view model public__stg.stg_order_details ..... [RUN]
11:17:52 2 of 7 OK created sql view model public__stg.stg_order_details ..... [CREATE VIEW in 0.12s]
11:17:52 3 of 7 START sql view model public__stg.stg_orders ..... [RUN]
11:17:52 3 of 7 OK created sql view model public__stg.stg_orders ..... [CREATE VIEW in 0.10s]
11:17:52 4 of 7 START sql view model public__stg.stg_products ..... [RUN]
11:17:53 4 of 7 OK created sql view model public__stg.stg_products ..... [CREATE VIEW in 0.08s]
11:17:53 5 of 7 START sql table model public__int.int_orders ..... [RUN]
11:17:53 5 of 7 OK created sql table model public__int.int_orders ..... [SELECT 431 in 0.11s]
11:17:53 6 of 7 START sql table model public__int.int_products ..... [RUN]
11:17:53 6 of 7 OK created sql table model public__int.int_products ..... [SELECT 4 in 0.11s]
11:17:53 7 of 7 START sql table model public__int.int_order_details ..... [RUN]
11:17:53 7 of 7 OK created sql table model public__int.int_order_details ..... [SELECT 431 in 0.11s]
11:17:53
11:17:53 Finished running 4 table models, 3 view models in 0 hours 0 minutes and 1.24 seconds (1.24s).
11:17:53
11:17:53 Completed successfully
11:17:53
11:17:53 Done. PASS=7 WARN=0 ERROR=0 SKIP=0 TOTAL=7
11:17:57 Running with dbt=1.8.5
11:17:57 Registered adapter: postgres=1.8.2
11:17:57 [WARNING]: Configuration paths exist in your dbt_project.yml file which do not apply to any resources.
There are 2 unused configuration paths:
- models.my_project.mart
- models.my_project._fct
11:17:57 Found 7 models, 33 data tests, 4 sources, 417 macros
11:17:57
11:17:58 Concurrency: 1 threads (target='dev')
11:17:58
11:17:58 1 of 33 START test not_null_int_order_details_order_detail_id ..... [RUN]
11:17:58 1 of 33 PASS not_null_int_order_details_order_detail_id ..... [PASS in 0.07s]
11:17:58 2 of 33 START test not_null_int_orders_order_id ..... [RUN]
11:17:58 2 of 33 PASS not_null_int_orders_order_id ..... [PASS in 0.05s]
11:17:58 3 of 33 START test not_null_int_products_brand_name ..... [RUN]
11:17:58 3 of 33 PASS not_null_int_products_brand_name ..... [PASS in 0.05s]
11:17:58 4 of 33 START test not_null_int_products_product_name ..... [RUN]
11:17:58 4 of 33 PASS not_null_int_products_product_name ..... [PASS in 0.04s]
11:17:58 5 of 33 START test not_null_stg_brands_brand_id ..... [RUN]
11:17:58 5 of 33 PASS not_null_stg_brands_brand_id ..... [PASS in 0.07s]
11:17:58 6 of 33 START test not_null_stg_order_details_order_detail_id ..... [RUN]
11:17:58 6 of 33 PASS not_null_stg_order_details_order_detail_id ..... [PASS in 0.06s]
11:17:58 7 of 33 START test not_null_stg_orders_order_id ..... [RUN]
11:17:58 7 of 33 PASS not_null_stg_orders_order_id ..... [PASS in 0.06s]
11:17:58 8 of 33 START test not_null_stg_products_brand_id ..... [RUN]
11:17:58 8 of 33 PASS not_null_stg_products_brand_id ..... [PASS in 0.05s]
11:17:58 9 of 33 START test not_null_stg_products_product_id ..... [RUN]
11:17:58 9 of 33 PASS not_null_stg_products_product_id ..... [PASS in 0.05s]
11:17:58 10 of 33 START test source_not_null_store_brands_brand_id ..... [RUN]
11:17:58 10 of 33 PASS source_not_null_store_brands_brand_id ..... [PASS in 0.04s]
11:17:58 11 of 33 START test source_not_null_store_brands_name ..... [RUN]
11:17:58 11 of 33 PASS source_not_null_store_brands_name ..... [PASS in 0.05s]
11:17:58 12 of 33 START test source_not_null_store_order_details_order_detail_id ..... [RUN]
11:17:58 12 of 33 PASS source_not_null_store_order_details_order_detail_id ..... [PASS in 0.05s]
11:17:58 13 of 33 START test source_not_null_store_order_details_price ..... [RUN]
11:17:58 13 of 33 PASS source_not_null_store_order_details_price ..... [PASS in 0.06s]
11:17:58 14 of 33 START test source_not_null_store_order_details_quantity ..... [RUN]
11:17:58 14 of 33 PASS source_not_null_store_order_details_quantity ..... [PASS in 0.05s]
11:17:58 15 of 33 START test source_not_null_store_orders_order_date ..... [RUN]
11:17:59 15 of 33 PASS source_not_null_store_orders_order_date ..... [PASS in 0.06s]
```

```
11:17:59 16 of 33 START test source_not_null_store_orders_order_id ..... [RUN]
11:17:59 16 of 33 PASS source_not_null_store_orders_order_id ..... [PASS in 0.05s]
11:17:59 17 of 33 START test source_not_null_store_products_name ..... [RUN]
11:17:59 17 of 33 PASS source_not_null_store_products_name ..... [PASS in 0.06s]
11:17:59 18 of 33 START test source_not_null_store_products_price ..... [RUN]
11:17:59 18 of 33 PASS source_not_null_store_products_price ..... [PASS in 0.05s]
11:17:59 19 of 33 START test source_not_null_store_products_product_id ..... [RUN]
11:17:59 19 of 33 PASS source_not_null_store_products_product_id ..... [PASS in 0.07s]
11:17:59 20 of 33 START test source_relationships_store_order_details_order_id_order_id_source_store_orders_ [RUN]
11:17:59 20 of 33 PASS source_relationships_store_order_details_order_id_order_id_source_store_orders_ [PASS in 0.06s]
11:17:59 21 of 33 START test source_relationships_store_order_details_product_id_product_id_source_store_products_ [RUN]
11:17:59 21 of 33 PASS source_relationships_store_order_details_product_id_product_id_source_store_products_ [PASS in 0.06s]
11:17:59 22 of 33 START test source_relationships_store_products_brand_id_brand_id_source_store_brands_ [RUN]
11:17:59 22 of 33 PASS source_relationships_store_products_brand_id_brand_id_source_store_brands_ [PASS in 0.07s]
11:17:59 23 of 33 START test source_unique_store_brands_brand_id ..... [RUN]
11:17:59 23 of 33 PASS source_unique_store_brands_brand_id ..... [PASS in 0.05s]
11:17:59 24 of 33 START test source_unique_store_order_details_order_detail_id ..... [RUN]
11:17:59 24 of 33 PASS source_unique_store_order_details_order_detail_id ..... [PASS in 0.07s]
11:17:59 25 of 33 START test source_unique_store_orders_order_id ..... [RUN]
11:17:59 25 of 33 PASS source_unique_store_orders_order_id ..... [PASS in 0.05s]
11:17:59 26 of 33 START test source_unique_store_products_product_id ..... [RUN]
11:17:59 26 of 33 PASS source_unique_store_products_product_id ..... [PASS in 0.07s]
11:17:59 27 of 33 START test unique_int_order_details_order_detail_id ..... [RUN]
11:17:59 27 of 33 PASS unique_int_order_details_order_detail_id ..... [PASS in 0.05s]
11:17:59 28 of 33 START test unique_int_orders_order_id ..... [RUN]
11:17:59 28 of 33 PASS unique_int_orders_order_id ..... [PASS in 0.05s]
11:17:59 29 of 33 START test unique_int_products_product_name ..... [RUN]
11:17:59 29 of 33 PASS unique_int_products_product_name ..... [PASS in 0.07s]
11:17:59 30 of 33 START test unique_stg_brands_brand_id ..... [RUN]
11:17:59 30 of 33 PASS unique_stg_brands_brand_id ..... [PASS in 0.05s]
11:17:59 31 of 33 START test unique_stg_order_details_order_detail_id ..... [RUN]
11:17:59 31 of 33 PASS unique_stg_order_details_order_detail_id ..... [PASS in 0.06s]
11:17:59 32 of 33 START test unique_stg_orders_order_id ..... [RUN]
11:18:00 32 of 33 PASS unique_stg_orders_order_id ..... [PASS in 0.07s]
11:18:00 33 of 33 START test unique_stg_products_product_id ..... [RUN]
11:18:00 33 of 33 PASS unique_stg_products_product_id ..... [PASS in 0.06s]
11:18:00 Finished running 33 data tests in 0 hours 0 minutes and 2.23 seconds (2.23s).
11:18:00 Completed successfully
11:18:00 Done. PASS=33 WARN=0 ERROR=0 SKIP=0 TOTAL=33
(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
o $
```

Berikut ini adalah hasil dokumentasi dari perintah “dbt run && dbt test”. Dari gambar ini menunjukkan bahwa dari perintah “dbt run” maupun “dbt test” telah berhasil dengan ditandai oleh keterangan “Completed successfully” dan tanpa error pada setiap hasil perintah “dbt run” maupun “dbt test”.

8. Lalu pada aplikasi DBeaver pilih connection dbname store dengan port 5431, lalu pilih Databases > store > Schemas.



Skema dengan nama “public__int” sudah berhasil dibuat.

9. Kemudian buat file dengan nama “fct_orders.sql” dan buat query seperti berikut ini :

```
my_project > models > _fct > fct_orders.sql
1  select
2      orders.*
3      , case
4          when orders.order_id % 2 != 0 then 'marketing'
5          else 'finance'
6      end as mart_flaging
7  from {{ ref('int_orders') }} as orders
```

10. Buat file dengan nama “fct_order_details.sql” dan buat query seperti berikut ini :

```
my_project > models > _fct > fct_order_details.sql
1  select *
2  from {{ ref('int_order_details') }}
```

11. Buat file dengan nama “mart_marketing_orders.sql” dan buat query seperti gambar di bawah ini :

```
my_project > models > mart > mart_marketing_orders.sql
1  select *
2  from {{ ref('fct_orders') }}
3  where mart_flaging = 'marketing'
```

12. Buat file dengan nama “mart_finance_orders.sql” dan buat query seperti gambar di bawah ini :

```
my_project > models > mart > mart_finance_orders.sql
1  select *
2  from {{ ref('fct_orders') }}
3  where mart_flaging = 'finance'
```

13. Lalu jalankan perintah “dbt run && dbt test” lagi untuk menguji apakah DBT dapat berjalan dengan baik.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
bash - my_project + - - - - - x

(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
$ dbt run && dbt test
13:07:01 Running with dbt=1.8.5
13:07:01 Registered adapter: postgres=1.8.2
13:07:01 Found 11 models, 33 data tests, 4 sources, 417 macros
13:07:01
13:07:02 Concurrency: 1 threads (target='dev')
13:07:02
```

```

13:07:02 1 of 11 START sql table model public__stg.stg_brands ..... [RUN]
13:07:02 1 of 11 OK created sql table model public__stg.stg_brands ..... [SELECT 3 in 0.15s]
13:07:02 2 of 11 START sql view model public__stg.stg_order_details ..... [RUN]
13:07:02 2 of 11 OK created sql view model public__stg.stg_order_details ..... [CREATE VIEW in 0.11s]
13:07:02 3 of 11 START sql view model public__stg.stg_orders ..... [RUN]
13:07:02 3 of 11 OK created sql view model public__stg.stg_orders ..... [CREATE VIEW in 0.08s]
13:07:02 4 of 11 START sql view model public__stg.stg_products ..... [RUN]
13:07:02 4 of 11 OK created sql view model public__stg.stg_products ..... [CREATE VIEW in 0.07s]
13:07:02 5 of 11 START sql table model public__int.int_orders ..... [RUN]
13:07:02 5 of 11 OK created sql table model public__int.int_orders ..... [SELECT 431 in 0.10s]
13:07:02 6 of 11 START sql table model public__int.int_products ..... [RUN]
13:07:02 6 of 11 OK created sql table model public__int.int_products ..... [SELECT 4 in 0.09s]
13:07:02 7 of 11 START sql table model public__fct.fct_orders ..... [RUN]
13:07:02 7 of 11 OK created sql table model public__fct.fct_orders ..... [SELECT 431 in 0.10s]
13:07:02 8 of 11 START sql table model public__int.int_order_details ..... [RUN]
13:07:02 8 of 11 OK created sql table model public__int.int_order_details ..... [SELECT 431 in 0.13s]
13:07:03 9 of 11 START sql table model public_mart.mart_finance_orders ..... [RUN]
13:07:03 9 of 11 OK created sql table model public_mart.mart_finance_orders ..... [SELECT 215 in 0.08s]
13:07:03 10 of 11 START sql table model public_mart.mart_marketing_orders ..... [RUN]
13:07:03 10 of 11 OK created sql table model public_mart.mart_marketing_orders ..... [SELECT 216 in 0.15s]
13:07:03 11 of 11 START sql table model public__fct.fct_order_details ..... [RUN]
13:07:03 11 of 11 OK created sql table model public__fct.fct_order_details ..... [SELECT 431 in 0.10s]
13:07:03
13:07:03 Finished running 8 table models, 3 view models in 0 hours 0 minutes and 1.74 seconds (1.74s).
13:07:03
13:07:03 Completed successfully
13:07:03
13:07:03 Done. PASS=11 WARN=0 ERROR=0 SKIP=0 TOTAL=11
13:07:11 Running with dbt=1.8.5
13:07:11 Registered adapter: postgres=1.8.2
13:07:11 Found 11 models, 33 data tests, 4 sources, 417 macros
13:07:11
13:07:12 Concurrency: 1 threads (target='dev')
13:07:12
13:07:12 1 of 33 START test not_null_int_order_details_order_detail_id ..... [RUN]
13:07:12 1 of 33 PASS not_null_int_order_details_order_detail_id ..... [PASS in 0.08s]
13:07:12 2 of 33 START test not_null_int_orders_order_id ..... [RUN]
13:07:12 2 of 33 PASS not_null_int_orders_order_id ..... [PASS in 0.06s]
13:07:12 3 of 33 START test not_null_int_products_brand_name ..... [RUN]
13:07:12 3 of 33 PASS not_null_int_products_brand_name ..... [PASS in 0.06s]
13:07:12 4 of 33 START test not_null_int_products_product_name ..... [RUN]
13:07:12 4 of 33 PASS not_null_int_products_product_name ..... [PASS in 0.04s]
13:07:12 5 of 33 START test not_null_stg_brands_brand_id ..... [RUN]
13:07:12 5 of 33 PASS not_null_stg_brands_brand_id ..... [PASS in 0.05s]
13:07:12 6 of 33 START test not_null_stg_order_details_order_detail_id ..... [RUN]
13:07:12 6 of 33 PASS not_null_stg_order_details_order_detail_id ..... [PASS in 0.06s]
13:07:12 7 of 33 START test not_null_stg_orders_order_id ..... [RUN]
13:07:12 7 of 33 PASS not_null_stg_orders_order_id ..... [PASS in 0.06s]
13:07:12 8 of 33 START test not_null_stg_products_brand_id ..... [RUN]
13:07:12 8 of 33 PASS not_null_stg_products_brand_id ..... [PASS in 0.05s]
13:07:12 9 of 33 START test not_null_stg_products_product_id ..... [RUN]
13:07:12 9 of 33 PASS not_null_stg_products_product_id ..... [PASS in 0.04s]
13:07:12 10 of 33 START test source_not_null_store_brands_brand_id ..... [RUN]
13:07:12 10 of 33 PASS source_not_null_store_brands_brand_id ..... [PASS in 0.07s]
13:07:12 11 of 33 START test source_not_null_store_brands_name ..... [RUN]
13:07:12 11 of 33 PASS source_not_null_store_brands_name ..... [PASS in 0.06s]
13:07:12 12 of 33 START test source_not_null_store_order_details_order_detail_id ..... [RUN]
13:07:12 12 of 33 PASS source_not_null_store_order_details_order_detail_id ..... [PASS in 0.05s]
13:07:12 13 of 33 START test source_not_null_store_order_details_price ..... [RUN]
13:07:12 13 of 33 PASS source_not_null_store_order_details_price ..... [PASS in 0.06s]
13:07:12 14 of 33 START test source_not_null_store_order_details_quantity ..... [RUN]
13:07:12 14 of 33 PASS source_not_null_store_order_details_quantity ..... [PASS in 0.05s]
13:07:12 15 of 33 START test source_not_null_store_orders_order_date ..... [RUN]
13:07:12 15 of 33 PASS source_not_null_store_orders_order_date ..... [PASS in 0.05s]
13:07:12 16 of 33 START test source_not_null_store_orders_order_id ..... [RUN]
13:07:12 16 of 33 PASS source_not_null_store_orders_order_id ..... [PASS in 0.04s]
13:07:13 17 of 33 START test source_not_null_store_products_name ..... [RUN]
13:07:13 17 of 33 PASS source_not_null_store_products_name ..... [PASS in 0.05s]
13:07:13 18 of 33 START test source_not_null_store_products_price ..... [RUN]
13:07:13 18 of 33 PASS source_not_null_store_products_price ..... [PASS in 0.04s]
13:07:13 19 of 33 START test source_not_null_store_products_product_id ..... [RUN]
13:07:13 19 of 33 PASS source_not_null_store_products_product_id ..... [PASS in 0.05s]
13:07:13 20 of 33 START test source_relationships_store_order_details_order_id_order_id_source_store_orders_ [RUN]
13:07:13 20 of 33 PASS source_relationships_store_order_details_order_id_order_id_source_store_orders_ [PASS in 0.06s]
13:07:13 21 of 33 START test source_relationships_store_order_details_product_id_product_id_source_store_products_ [RUN]
13:07:13 21 of 33 PASS source_relationships_store_order_details_product_id_product_id_source_store_products_ [PASS in 0.06s]

```



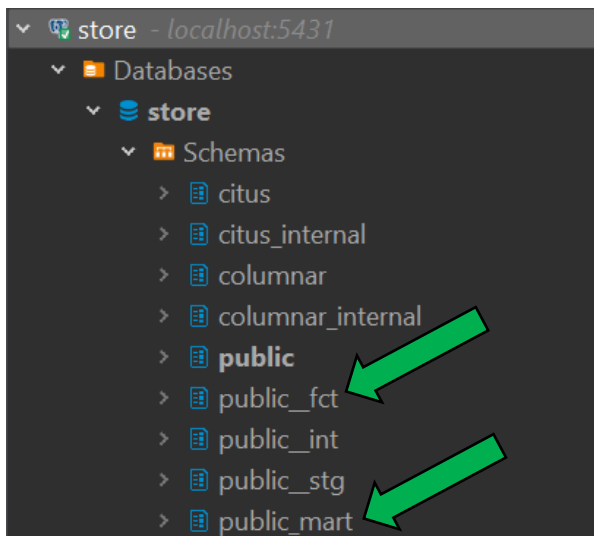
```

13:07:13 22 of 33 START test source_relationships_store_products_brand_id_brand_id_source_store_brands_ [RUN]
13:07:13 22 of 33 PASS source_relationships_store_products_brand_id_brand_id_source_store_brands_ [PASS in 0.05s]
13:07:13 23 of 33 START test source_unique_store_brands_brand_id ..... [RUN]
13:07:13 23 of 33 PASS source_unique_store_brands_brand_id ..... [PASS in 0.05s]
13:07:13 24 of 33 START test source_unique_store_order_details_order_detail_id ..... [RUN]
13:07:13 24 of 33 PASS source_unique_store_order_details_order_detail_id ..... [PASS in 0.07s]
13:07:13 25 of 33 START test source_unique_store_orders_order_id ..... [RUN]
13:07:13 25 of 33 PASS source_unique_store_orders_order_id ..... [PASS in 0.05s]
13:07:13 26 of 33 START test source_unique_store_products_product_id ..... [RUN]
13:07:13 26 of 33 PASS source_unique_store_products_product_id ..... [PASS in 0.08s]
13:07:13 27 of 33 START test unique_int_order_details_order_detail_id ..... [RUN]
13:07:13 27 of 33 PASS unique_int_order_details_order_detail_id ..... [PASS in 0.06s]
13:07:13 28 of 33 START test unique_int_orders_order_id ..... [RUN]
13:07:13 28 of 33 PASS unique_int_orders_order_id ..... [PASS in 0.04s]
13:07:13 29 of 33 START test unique_int_products_product_name ..... [RUN]
13:07:13 29 of 33 PASS unique_int_products_product_name ..... [PASS in 0.06s]
13:07:13 30 of 33 START test unique_stg_brands_brand_id ..... [RUN]
13:07:13 30 of 33 PASS unique_stg_brands_brand_id ..... [PASS in 0.06s]
13:07:13 31 of 33 START test unique_stg_order_details_order_detail_id ..... [RUN]
13:07:13 31 of 33 PASS unique_stg_order_details_order_detail_id ..... [PASS in 0.05s]
13:07:13 32 of 33 START test unique_stg_orders_order_id ..... [RUN]
13:07:13 32 of 33 PASS unique_stg_orders_order_id ..... [PASS in 0.06s]
13:07:13 33 of 33 START test unique_stg_products_product_id ..... [RUN]
13:07:14 33 of 33 PASS unique_stg_products_product_id ..... [PASS in 0.05s]
13:07:14 Finished running 33 data tests in 0 hours 0 minutes and 2.31 seconds (2.31s).
13:07:14 Completed successfully
13:07:14 Done. PASS=33 WARN=0 ERROR=0 SKIP=0 TOTAL=33
(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
$

```

Berikut ini adalah hasil dokumentasi dari perintah “dbt run && dbt test”. Dari gambar ini menunjukkan bahwa dari perintah “dbt run” maupun “dbt test” telah berhasil dengan ditandai oleh keterangan “**Completed successfully**” dan tanpa error pada setiap hasil perintah “dbt run” maupun “dbt test”.

14. Lalu pada aplikasi DBeaver pilih connection dbname store dengan port 5431, lalu pilih Databases > store > Schemas.



Skema dengan nama “public__fct” dan “public__mart” sudah berhasil dibuat.

15. Kemudian kita pergi ke webpage dbt packages dan copy packages untuk dbt utils dan dbt expectations tersebut. Lalu buat file dengan nama “packages.yml” dan paste packages yang sudah di copy sebelumnya ke dalam file tersebut.

```

my_project > ! packages.yml
1  packages:
2    - package: dbt-labs/dbt_utils
3      version: 1.2.0
4    - package: calogica/dbt_expectations
5      version: 0.10.3

```

Berikut ini adalah tampilan isi dari file dengan nama “packages.yml” yang sudah dibuat sebelumnya.

16. Buat dependensi ke packages yang sudah kita dapat dengan melakukan perintah “dbt deps”.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
bash - my_project + - - - - -

tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
$ dbt deps
15:47:35 Running with dbt=1.8.5
15:47:36 Updating lock file in file path: C:\Users\tawhe\Documents\DBT-DEMO\my_project\package-lock.yml
15:47:36 Installing dbt-labs/dbt_utils
15:47:37 Installed from version 1.2.0
15:47:37 Up to date!
15:47:37 Installing calogica/dbt_expectations
15:47:44 Installed from version 0.10.3
15:47:44 Up to date!
15:47:44 Installing calogica/dbt_date
15:47:45 Installed from version 0.10.1
15:47:45 Up to date!
(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)

```

Berikut adalah hasil dari perintah “dbt deps”, dan untuk packages dbt utils sudah bisa digunakan.

17. Buat file dengan nama “fct_schema.yml” dan buat query seperti gambar di bawah ini :

```

my_project > models > _fct > ! fct_schema.yml
1  version: 2
2
3  models:
4    - name: fct_orders
5      columns:
6        - name: order_id
7          tests:
8            - not_null
9            - unique
10         tests:
11           - dbt_expectations.expect_table_row_count_to_equal_other_table:
12             compare_model: ref("int_order_details")
13
14     - name: fct_order_details
15       columns:
16         - name: order_detail_id
17           tests:
18             - not_null
19             - unique
20         tests:
21           - dbt_utils.unique_combination_of_columns:
22             combination_of_columns:
23               - order_id
24               - product_id

```

18. Kemudian jalankan perintah “dbt run && dbt test” untuk menguji packages yang sudah kita gunakan.


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
bash - my_project + - - - - - x

(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
$ dbt run && dbt test
15:50:23 Running with dbt=1.8.5
15:50:23 Registered adapter: postgres=1.8.2
15:50:24 Found 11 models, 39 data tests, 4 sources, 800 macros
15:50:24 Concurrency: 1 threads (target='dev')
15:50:24
15:50:24 1 of 11 START sql table model public_stg.stg_brands ..... [RUN]
15:50:24 1 of 11 OK created sql table model public_stg.stg_brands ..... [SELECT 3 in 0.17s]
15:50:24 2 of 11 START sql view model public_stg.stg_order_details ..... [RUN]
15:50:24 2 of 11 OK created sql view model public_stg.stg_order_details ..... [CREATE VIEW in 0.10s]
15:50:24 3 of 11 START sql view model public_stg.stg_orders ..... [RUN]
15:50:25 3 of 11 OK created sql view model public_stg.stg_orders ..... [CREATE VIEW in 0.10s]
15:50:25 4 of 11 START sql view model public_stg.stg_products ..... [RUN]
15:50:25 4 of 11 OK created sql view model public_stg.stg_products ..... [CREATE VIEW in 0.10s]
15:50:25 5 of 11 START sql table model public_int.int_orders ..... [RUN]
15:50:25 5 of 11 OK created sql table model public_int.int_orders ..... [SELECT 431 in 0.09s]
15:50:25 6 of 11 START sql table model public_int.int_products ..... [RUN]
15:50:25 6 of 11 OK created sql table model public_int.int_products ..... [SELECT 4 in 0.09s]
15:50:25 7 of 11 START sql table model public_fct.fct_orders ..... [RUN]
15:50:25 7 of 11 OK created sql table model public_fct.fct_orders ..... [SELECT 431 in 0.15s]
15:50:25 8 of 11 START sql table model public_int.int_order_details ..... [RUN]
15:50:25 8 of 11 OK created sql table model public_int.int_order_details ..... [SELECT 431 in 0.11s]
15:50:25 9 of 11 START sql table model public_mart.mart_finance_orders ..... [RUN]
15:50:25 9 of 11 OK created sql table model public_mart.mart_finance_orders ..... [SELECT 215 in 0.10s]
15:50:25 10 of 11 START sql table model public_mart.mart_marketing_orders ..... [RUN]
15:50:25 10 of 11 OK created sql table model public_mart.mart_marketing_orders ..... [SELECT 216 in 0.12s]
15:50:25 11 of 11 START sql table model public_fct.fct_order_details ..... [RUN]
15:50:25 11 of 11 OK created sql table model public_fct.fct_order_details ..... [SELECT 431 in 0.10s]
15:50:25
15:50:25 Finished running 8 table models, 3 view models in 0 hours 0 minutes and 1.81 seconds (1.81s).
15:50:26
15:50:26 Completed successfully
15:50:26
15:50:26 Done. PASS=11 WARN=0 ERROR=0 SKIP=0 TOTAL=11
15:50:29 Running with dbt=1.8.5
15:50:30 Registered adapter: postgres=1.8.2
15:50:30 Found 11 models, 39 data tests, 4 sources, 800 macros
15:50:30 Concurrency: 1 threads (target='dev')
15:50:30
15:50:30 1 of 39 START test dbt_expectations_expect_table_row_count_to_equal_other_table_fct_orders_ref_fct_order_details_
[RUN]
15:50:31 1 of 39 PASS dbt_expectations_expect_table_row_count_to_equal_other_table_fct_orders_ref_fct_order_details_ [PAS
S in 0.10s]
15:50:31 2 of 39 START test dbt_utils_unique_combination_of_columns_fct_order_details_order_id_product_id [RUN]
15:50:31 2 of 39 PASS dbt_utils_unique_combination_of_columns_fct_order_details_order_id_product_id [PASS in 0.05s]
15:50:31 3 of 39 START test not_null_fct_order_details_order_detail_id ..... [RUN]
15:50:31 3 of 39 PASS not_null_fct_order_details_order_detail_id ..... [PASS in 0.07s]
15:50:31 4 of 39 START test not_null_fct_orders_order_id ..... [RUN]
15:50:31 4 of 39 PASS not_null_fct_orders_order_id ..... [PASS in 0.04s]
15:50:31 5 of 39 START test not_null_int_order_details_order_detail_id ..... [RUN]
15:50:31 5 of 39 PASS not_null_int_order_details_order_detail_id ..... [PASS in 0.06s]
15:50:31 6 of 39 START test not_null_int_orders_order_id ..... [RUN]
15:50:31 6 of 39 PASS not_null_int_orders_order_id ..... [PASS in 0.06s]
15:50:31 7 of 39 START test not_null_int_products_brand_name ..... [RUN]
15:50:31 7 of 39 PASS not_null_int_products_brand_name ..... [PASS in 0.06s]
15:50:31 8 of 39 START test not_null_int_products_product_name ..... [RUN]
15:50:31 8 of 39 PASS not_null_int_products_product_name ..... [PASS in 0.06s]
15:50:31 9 of 39 START test not_null_stg_brands_brand_id ..... [RUN]
15:50:31 9 of 39 PASS not_null_stg_brands_brand_id ..... [PASS in 0.06s]
15:50:31 10 of 39 START test not_null_stg_order_details_order_detail_id ..... [RUN]
15:50:31 10 of 39 PASS not_null_stg_order_details_order_detail_id ..... [PASS in 0.06s]
15:50:31 11 of 39 START test not_null_stg_orders_order_id ..... [RUN]
15:50:31 11 of 39 PASS not_null_stg_orders_order_id ..... [PASS in 0.04s]
15:50:31 12 of 39 START test not_null_stg_products_brand_id ..... [RUN]
15:50:31 12 of 39 PASS not_null_stg_products_brand_id ..... [PASS in 0.06s]
15:50:31 13 of 39 START test not_null_stg_products_product_id ..... [RUN]
15:50:31 13 of 39 PASS not_null_stg_products_product_id ..... [PASS in 0.06s]
15:50:31 14 of 39 START test source_not_null_store_brands_brand_id ..... [RUN]
15:50:31 14 of 39 PASS source_not_null_store_brands_brand_id ..... [PASS in 0.06s]
15:50:31 15 of 39 START test source_not_null_store_brands_name ..... [RUN]
15:50:31 15 of 39 PASS source_not_null_store_brands_name ..... [PASS in 0.06s]
15:50:31 16 of 39 START test source_not_null_store_order_details_order_detail_id ..... [RUN]
```

[illegible]

[illegible]

```
15:50:33 38 of 39 START test unique_stg_orders_order_id ..... [RUN]
15:50:33 38 of 39 PASS unique_stg_orders_order_id ..... [PASS in 0.05s]
15:50:33 39 of 39 START test unique_stg_products_product_id ..... [RUN]
15:50:33 39 of 39 PASS unique_stg_products_product_id ..... [PASS in 0.05s]
15:50:33
15:50:33 Finished running 39 data tests in 0 hours 0 minutes and 2.74 seconds (2.74s).
15:50:33
15:50:33 Completed successfully
15:50:33 38 of 39 START test unique_stg_orders_order_id ..... [RUN]
15:50:33 38 of 39 PASS unique_stg_orders_order_id ..... [PASS in 0.05s]
15:50:33 39 of 39 START test unique_stg_products_product_id ..... [RUN]
15:50:33 39 of 39 PASS unique_stg_products_product_id ..... [PASS in 0.05s]
15:50:33
15:50:33 Finished running 39 data tests in 0 hours 0 minutes and 2.74 seconds (2.74s).
15:50:33
15:50:33 Completed successfully
15:50:33
15:50:33 Done. PASS=39 WARN=0 ERROR=0 SKIP=0 TOTAL=39
15:50:33 Completed successfully
15:50:33
15:50:33 Done. PASS=39 WARN=0 ERROR=0 SKIP=0 TOTAL=39
15:50:33
15:50:33 Done. PASS=39 WARN=0 ERROR=0 SKIP=0 TOTAL=39
15:50:33 Done. PASS=39 WARN=0 ERROR=0 SKIP=0 TOTAL=39
(.venv)
tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
$
```

Berikut ini adalah hasil dokumentasi dari perintah “dbt run && dbt test”. Dari gambar ini menunjukkan bahwa dari perintah “dbt run” maupun “dbt test” telah berhasil dengan ditandai oleh keterangan “Completed successfully” dan tanpa error pada setiap hasil perintah “dbt run” maupun “dbt test”.

19. Buat file di folder “macros” dengan nama file “phone_normalization.sql”, lalu buat query sesuai dengan gambar di bawah ini :

```
my_project > macros > phone_normalization.sql
1  {% macro normalize_phone_number(column_name) %}
2    |    ltrim({{ column_name }}, '+')
3  {% endmacro %}
```

20. Kemudian buka file dengan nama “int_order_details.sql” pada folder “_int” dan tambahkan query “, {{ normalize_phone_number('orders.customer_phone') }} as normalized_phone” sesuai gambar di bawah ini :

```
my_project > models > _int > int_order_details.sql
1  select
2      details.order_detail_id
3      , details.order_id
4      , orders.order_at
5      , orders.customer_phone
6      , {{ normalize_phone_number('orders.customer_phone') }} as normalized_phone
7      , products.brand_id
8      , products.brand_name
9      , details.product_id
10     , products.product_name
11     , products.product_price
12     , details.order_qty
13     , details.unit_sales
14 from {{ ref('stg_order_details') }} as details
15 left join {{ ref('int_orders') }} as orders
16     on details.order_id = orders.order_id
17 left join {{ ref('int_products') }} as products
18     on details.product_id = products.product_id
```

Fungsinya adalah untuk menormalisasi nomor telfon agar dihilangkan tanda “+” nya hilang, dan semua nomor telfon menjadi tanpa tanda “+” dan sama.

21. Lalu buat folder dengan nama “materialization” kemudian di dalam folder tersebut buat lagi folder “models”, setelah itu buat file dengan nama “table.sql” pada folder “models” tersebut. Copy isi query yang ada di repository [dbt-demo/README.md at main · Immersive-DataEngineer-Resource/dbt-demo \(github.com\)](https://github.com/Immersive-DataEngineer-Resource/dbt-demo) pada bagian macro dan paste ke dalam file dengan nama “table.sql”.

```
my_project > macros > materializations > models > table.sql
1  {% materialization citus_materialization, adapter='postgres' %}
2
3      -- NOTE: For CITUS, We need to add distribution_column parameter
4      {% set distribution_column = config.get('distribution_column', default='id') -%}
5
6      {% set existing_relation = load_cached_relation(this) -%}
7      {% set target_relation = this.incorporate(type='table') %}
8      {% set intermediate_relation = make_intermediate_relation(target_relation) -%}
9      -- the intermediate_relation should not already exist in the database; get_relation
10     -- will return None in that case. Otherwise, we get a relation that we can drop
11     -- Later, before we try to use this name for the current operation
12     {% set preexisting_intermediate_relation = load_cached_relation(intermediate_relation) -%}
13     /*
14     See ../view/view.sql for more information about this relation.
15     */
16     {% set backup_relation_type = 'table' if existing_relation is none else existing_relation.type -%}
17     {% set backup_relation = make_backup_relation(target_relation, backup_relation_type) -%}
18     -- as above, the backup_relation should not already exist
19     {% set preexisting_backup_relation = load_cached_relation(backup_relation) -%}
20     -- grab current tables grants config for comparison later on
21     {% set grant_config = config.get('grants') %}
22
23     -- drop the temp relations if they exist already in the database
24     {{ drop_relation_if_exists(preexisting_intermediate_relation) }}
25     {{ drop_relation_if_exists(preexisting_backup_relation) }}
26
27     {{ run_hooks(pre_hooks, inside_transaction=False) }}
28
29     -- `BEGIN` happens here:
30     {{ run_hooks(pre_hooks, inside_transaction=True) }}
31
32     -- build model
33     {% call statement('main') -%}
34         {{ get_create_table_as_sql(False, intermediate_relation, sql) }}
35         -- NOTE: For CITUS, We need to turn the table into distributed table
36         select create_distributed_table('{{ intermediate_relation }}', '{{ distribution_column }}');
37     {% endcall %}
38
39     -- cleanup
40     {% if existing_relation is not none %}
41         /* Do the equivalent of rename_if_exists. 'existing_relation' could have been dropped
42         since the variable was first set. */
43         {% set existing_relation = load_cached_relation(existing_relation) %}
44         {% if existing_relation is not none %}
45             {{ adapter.rename_relation(existing_relation, backup_relation) }}
46         {% endif %}
47     {% endif %}
48
49     {{ adapter.rename_relation(intermediate_relation, target_relation) }}
50
51     {% do create_indexes(target_relation) %}
52
53     {{ run_hooks(post_hooks, inside_transaction=True) }}
54
55     {% set should_revoke = should_revoke(existing_relation, full_refresh_mode=True) %}
56     {% do apply_grants(target_relation, grant_config, should_revoke=should_revoke) %}
57
58     {% do persist_docs(target_relation, model) %}
59
60     -- `COMMIT` happens here
61     {{ adapter.commit() }}
62
63     -- finally, drop the existing/backup relation after the commit
64     {{ drop_relation_if_exists(backup_relation) }}
65
66     {{ run_hooks(post_hooks, inside_transaction=False) }}
67
68     {{ return({'relations': [target_relation]}) }}
69 {% endmaterialization %}
```

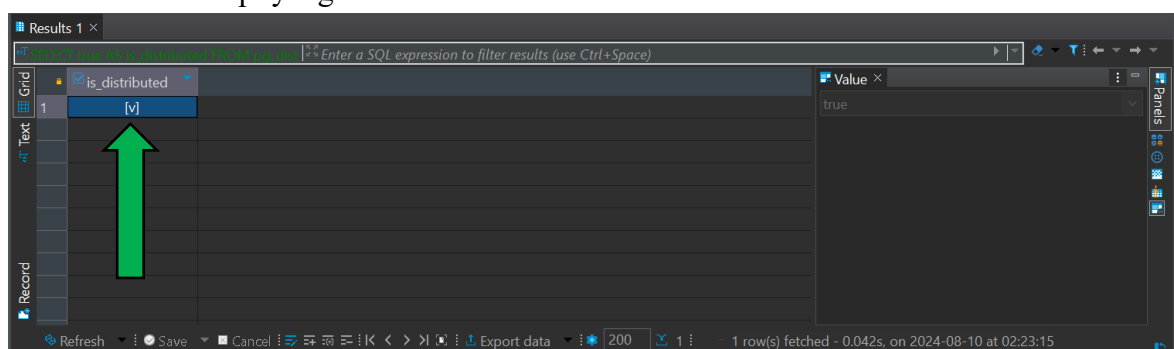
22. Buka folder “_stg”, lalu buat file dengan nama “stg_materialization_table.sql”. Kemudian copy isi query yang ada di repository [dbt-demo/README.md at main · Immersive-DataEngineer-Resource/dbt-demo \(github.com\)](https://github.com/Immersive-DataEngineer-Resource/dbt-demo) pada bagian macro dan paste ke dalam file dengan nama “stg_materialization_table.sql”.

```
my_project > models > _stg > stg_materialization_table.sql
1  {{
2  config(
3      materialized='citus_materialization',
4      distribution_column='order_date'
5  )
6  }}
7
8  with base as (
9      select
10         date(orders.order_date) as order_date
11         , order_details.quantity
12         , order_details.price
13     from {{ source('store', 'orders') }} AS orders
14     inner join {{ source('store', 'order_details') }} AS order_details
15         on orders.order_id = order_details.order_id
16 )
17
18 , aggregated_sales as (
19     select
20         order_date
21         , sum(quantity) as total_quantity
22         , sum(price) as total_revenue
23     from base
24     group by order_date
25 )
26
27 select *
28 from aggregated_sales
29 order by order_date
```

23. Buka aplikasi DBeaver kemudian pilih public__stg lalu open script dan buat “new script”, kemudian buat query seperti gambar berikut ini :

```
SELECT
    true AS is_distributed
FROM
    pg_dist_partition
WHERE
    logicalrelid = 'public__stg.stg_materialization_table'::regclass;
```

24. Lalu eksekusi script yang sudah dibuat.



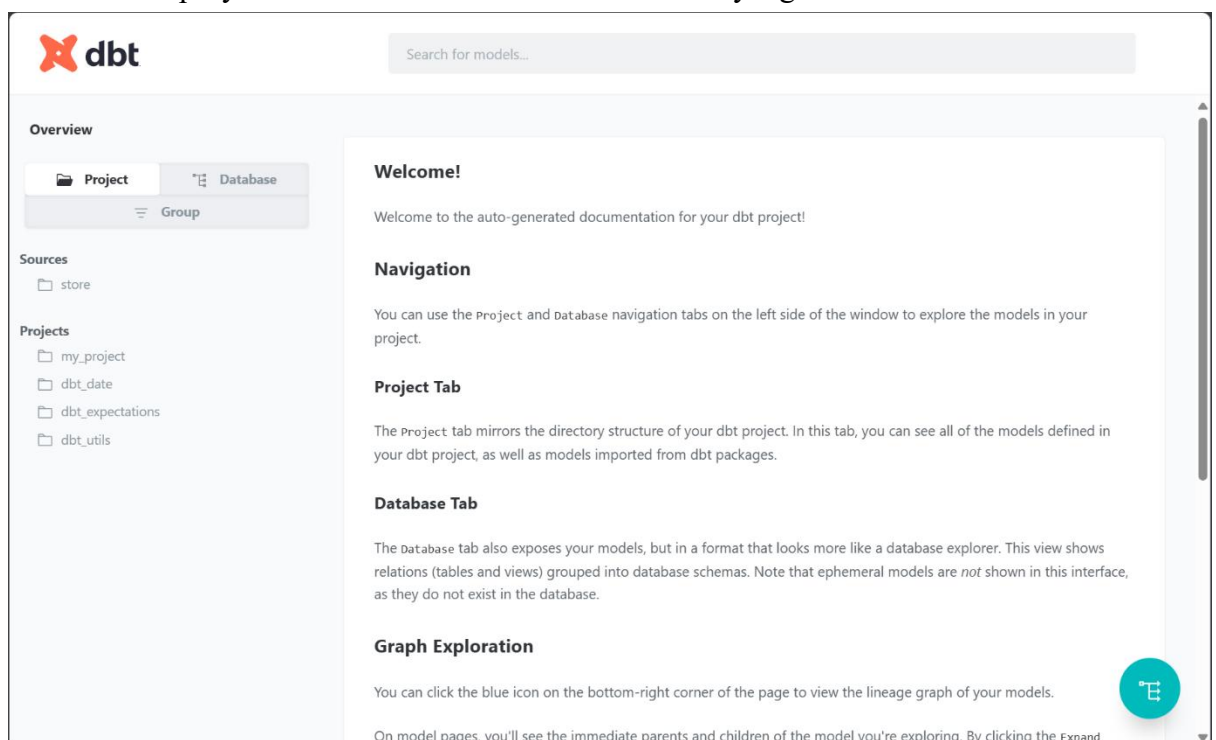
Query berhasil, dapat dibuktikan dari hasil centang pada results 1.

25. Selanjutnya lakukan perintah “dbt docs generate” untuk men generate seluruh dokumen yang sudah dibuat.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
bash - my_project + - - - ^ X

tawhe@Ibnu_Emil MINGW64 ~/Documents/DBT-DEMO/my_project (main)
$ dbt docs generate
19:28:54 Running with dbt=1.8.5
19:28:54 Registered adapter: postgres=1.8.2
19:28:55 Found 12 models, 40 data tests, 4 sources, 802 macros
19:28:55
19:28:55 Concurrency: 1 threads (target='dev')
19:28:55
19:28:56 Building catalog
19:28:56 Catalog written to C:\Users\tawhe\Documents\DBT-DEMO\my_project\target\catalog.json
(.venv)
```

26. Berikutnya lakukan perintah “dbt docs serve” untuk melayani dan menampilkan dokumentasi proyek DBT dalam bentuk antarmuka web yang sudah dibuat.



Perintah “dbt docs serve” sudah berhasil, dan proyek yang sudah dibuat telah disajikan dalam bentuk website.

27. Untuk tugas tambahannya, yaitu membuat kolom country sebagai normalisasi untuk nomor telfon jika angka nomor telfon diawali oleh angka “91”, maka isi dari kolom country tersebut adalah India, dan jika angka nomor telfon diawali oleh angka “62”, maka isi dari kolom country tersebut adalah Indonesia. Maka dari itu, untuk menjalankan syarat ini, yang harus dilakukan adalah dengan menambahkan quary seperti berikut ke dalam file “int_order_details.sql”:

```
“, case
  when {{ normalize_phone_number('orders.customer_phone') }} like '62%' then
    'Indonesia'
  else 'India'
end as country”
```

```

my_project > models > _int > int_order_details.sql
1  select
2      details.order_detail_id
3      , details.order_id
4      , orders.order_at
5      , orders.customer_phone
6      , {{ normalize_phone_number('orders.customer_phone') }} as normalized_phone
7      , case
8          when {{ normalize_phone_number('orders.customer_phone') }} like '62%' then 'Indonesia'
9          else 'India'
10         end as country
11      , products.brand_id
12      , products.brand_name
13      , details.product_id
14      , products.product_name
15      , products.product_price
16      , details.order_qty
17      , details.unit_sales
18  from {{ ref('stg_order_details') }} as details
19  left join {{ ref('int_orders') }} as orders
20      on details.order_id = orders.order_id
21  left join {{ ref('int_products') }} as products
22      on details.product_id = products.product_id

```

Berikut ini adalah tampilan query pada visual studio code saat implementasi querynya.

28. Langkah terakhir, yaitu jalankan query yang sudah dibuat dengan perintah “dbt run && dbt test”. Lalu hasil akan muncul pada aplikasi DBeaver di bagian connection dbname store dengan port 5431, lalu pilih Databases > store > Schemas > public__int > Tables > int_order_details.sql.

	customer_phone	normalized_phone	country	brand_id	brand_name	product_id
66	916622077325	916622077325	India	1	Apple	
67	916622077325	916622077325	India	3	Sony	
68	916622077325	916622077325	India	1	Apple	
69	+919979994193	919979994193	India	1	Apple	
70	+919979994193	919979994193	India	2	Samsung	
71	+919979994193	919979994193	India	1	Apple	
72	+919979994193	919979994193	India	1	Apple	
73	+919979994193	919979994193	India	3	Sony	
74	+919979994193	919979994193	India	1	Apple	
75	+919979994193	919979994193	India	2	Samsung	
76	+919979994193	919979994193	India	2	Samsung	
77	+919979994193	919979994193	India	3	Sony	
78	+919979994193	919979994193	India	1	Apple	
79	+627595362929	627595362929	Indonesia	1	Apple	
80	+627595362929	627595362929	Indonesia	2	Samsung	
81	+627595362929	627595362929	Indonesia	1	Apple	
82	+627595362929	627595362929	Indonesia	2	Samsung	
83	+627595362929	627595362929	Indonesia	2	Samsung	
84	+627595362929	627595362929	Indonesia	1	Apple	
85	+627595362929	627595362929	Indonesia	2	Samsung	
86	+627595362929	627595362929	Indonesia	2	Samsung	
87	+627595362929	627595362929	Indonesia	1	Apple	
88	+627595362929	627595362929	Indonesia	2	Samsung	
89	+627595362929	627595362929	Indonesia	1	Apple	
90	+627595362929	627595362929	Indonesia	3	Sony	
91	+627595362929	627595362929	Indonesia	1	Apple	
92	+627595362929	627595362929	Indonesia	1	Apple	
93	+627595362929	627595362929	Indonesia	1	Apple	
94	+627595362929	627595362929	Indonesia	2	Samsung	

Dari hasil diatas dapat disimpulkan bahwa query sudah **berhasil** dan **berjalan sesuai yang diharapkan**.