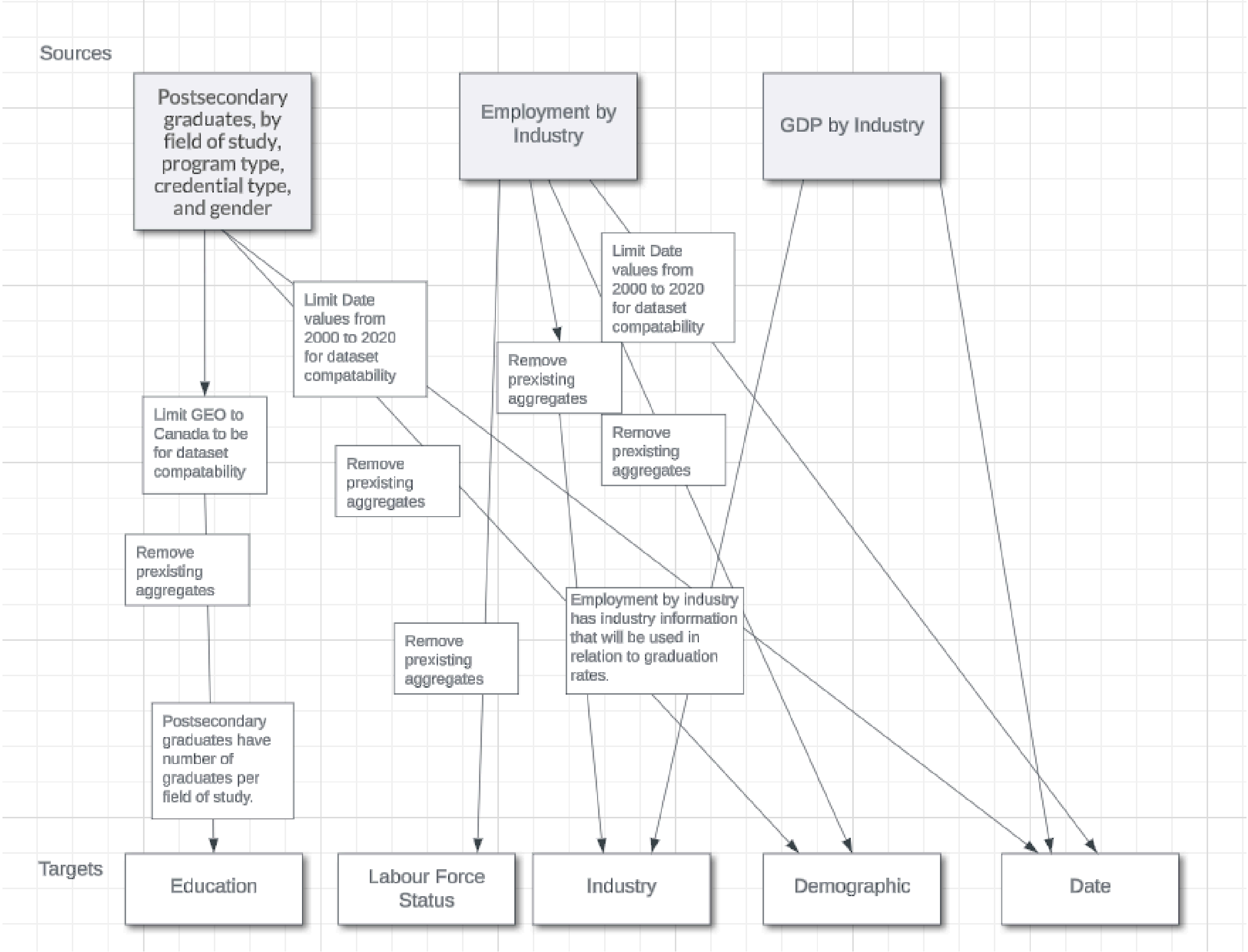
Database link is in data_staging.ipynb file and can be accessed either using the notebook or by using software like pgadmin

Github Link: https://github.com/KhaledElbasiouni/CSI4142-Project

High Level Data Staging Plan


Sources
Postsecondary graduates, by field of study, program type, credential type, and gender
Employment by Industry
GDP by Industry
Limit Date values from 2000 to 2020 for dataset compatability
Limit Date values from 2000 to 2020 for dataset compatability
Remove prexisting aggregates
Limit GEO to Canada to be for dataset compatability
Remove prexisting aggregates
Remove prexisting aggregates
Remove prexisting aggregates
Remove prexisting aggregates
Employment by industry has industry information that will be used in relation to graduation rates.
Postsecondary graduates have number of graduates per field of study.
Targets
Education
Labour Force Status
Industry
Demographic
Date

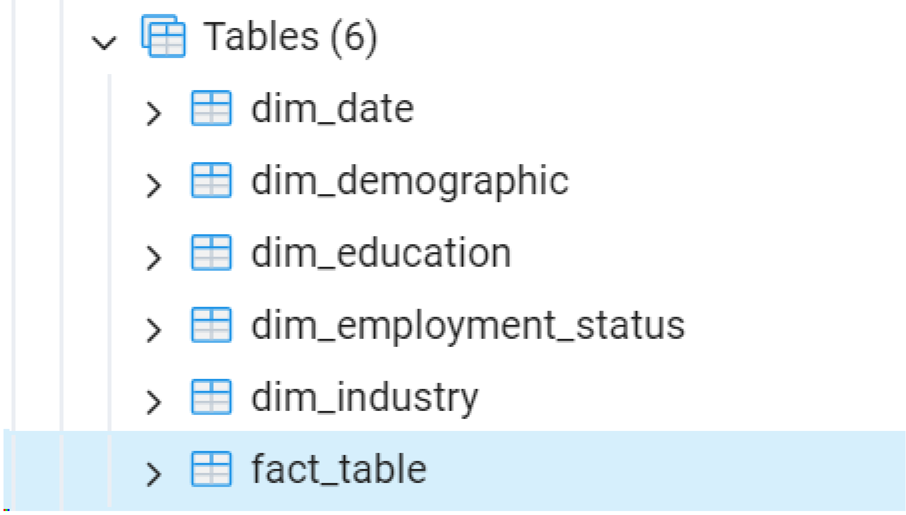
Data Quality Issues:

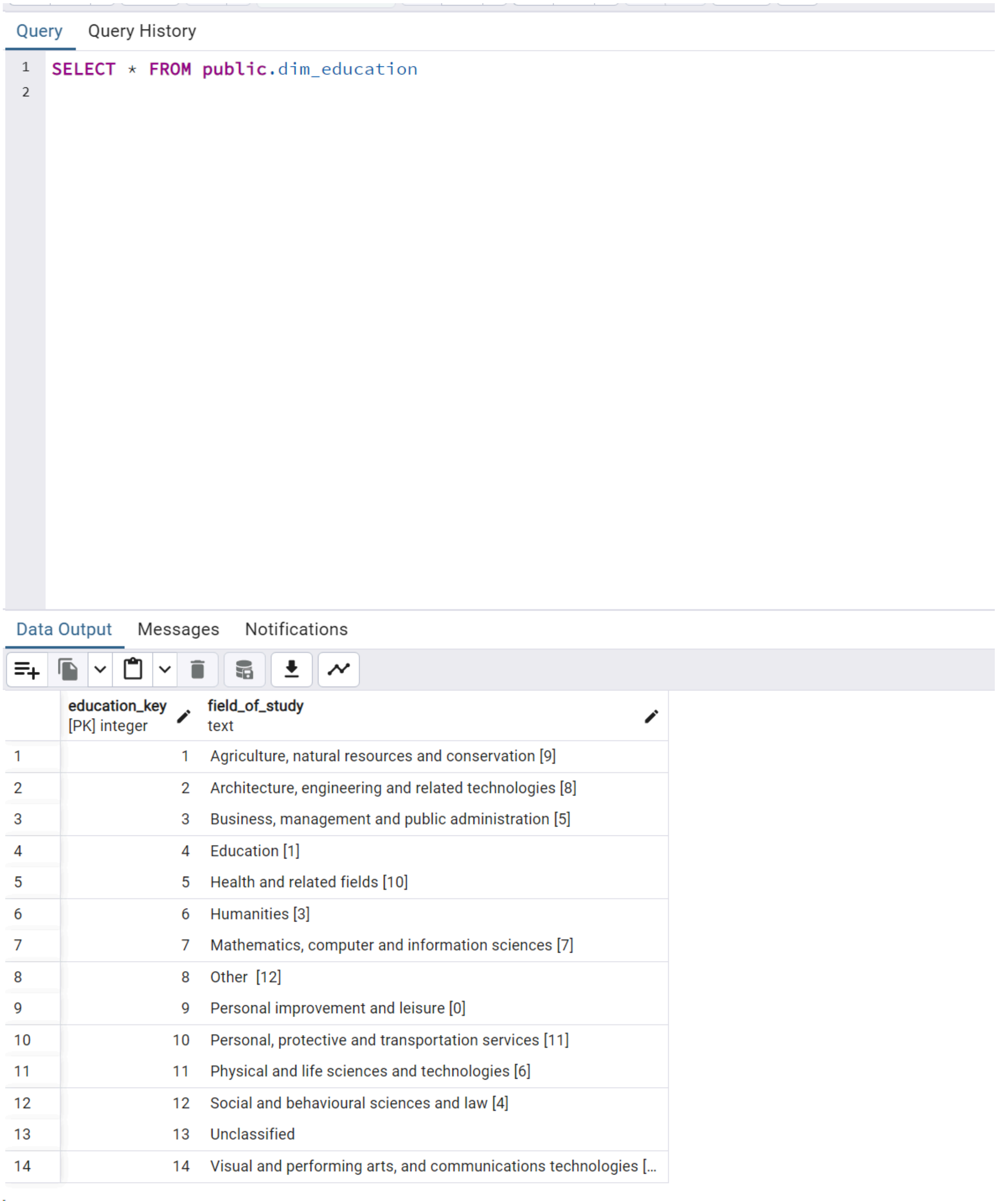
- Some data sets had different year ranges, so we had to find an appropriate intersection to be consistent among sources. We decided to use the year range 2000 - 2020.
- We found a new data set that serves our purposes better. Before, we had a dataset that did not provide us with unemployment information per industry. We now have a dataset that has a “North American Industry Classification System (NAICS)” column that better aligns with the NAICS column we have in a different dataset.
- We had to aggregate a lot of columns together to clean the dataframes to get them ready for use.
- We dropped many unnecessary columns. (Example -> ‘GEO’ column with 1 unique value: ‘Canada’)

- Renamed some columns/ low-cardinality column values for consistency across all datasets (Example: ‘Sex’ -> ‘Gender’ and ‘Man’, ‘Woman’ -> ‘Male’, ‘Female’)
- Removed some pre-existing aggregates and went for the lowest granular level (Example: ‘Total Gender’ -> ‘Male’ and ‘Female’)
- Made sure there aren’t any missing values across our datasets
- Made sure our low-cardinality column values contain no duplicates
- Converted some column data types into their ideal data type
- Mapped NAICS from different columns to contain exactly equal values between the 2 datasets with NAICS columns.
- Some NaN values were present in the employment per industry dataset. We decided to replace them with averages for those in the same group.
  For example:
  If we have a row

Date	Labour Force Characteristics	NAICS	Gender	Age Group	Value
2020	Unemployed	Utilities [22]	Male	55 years and over	0

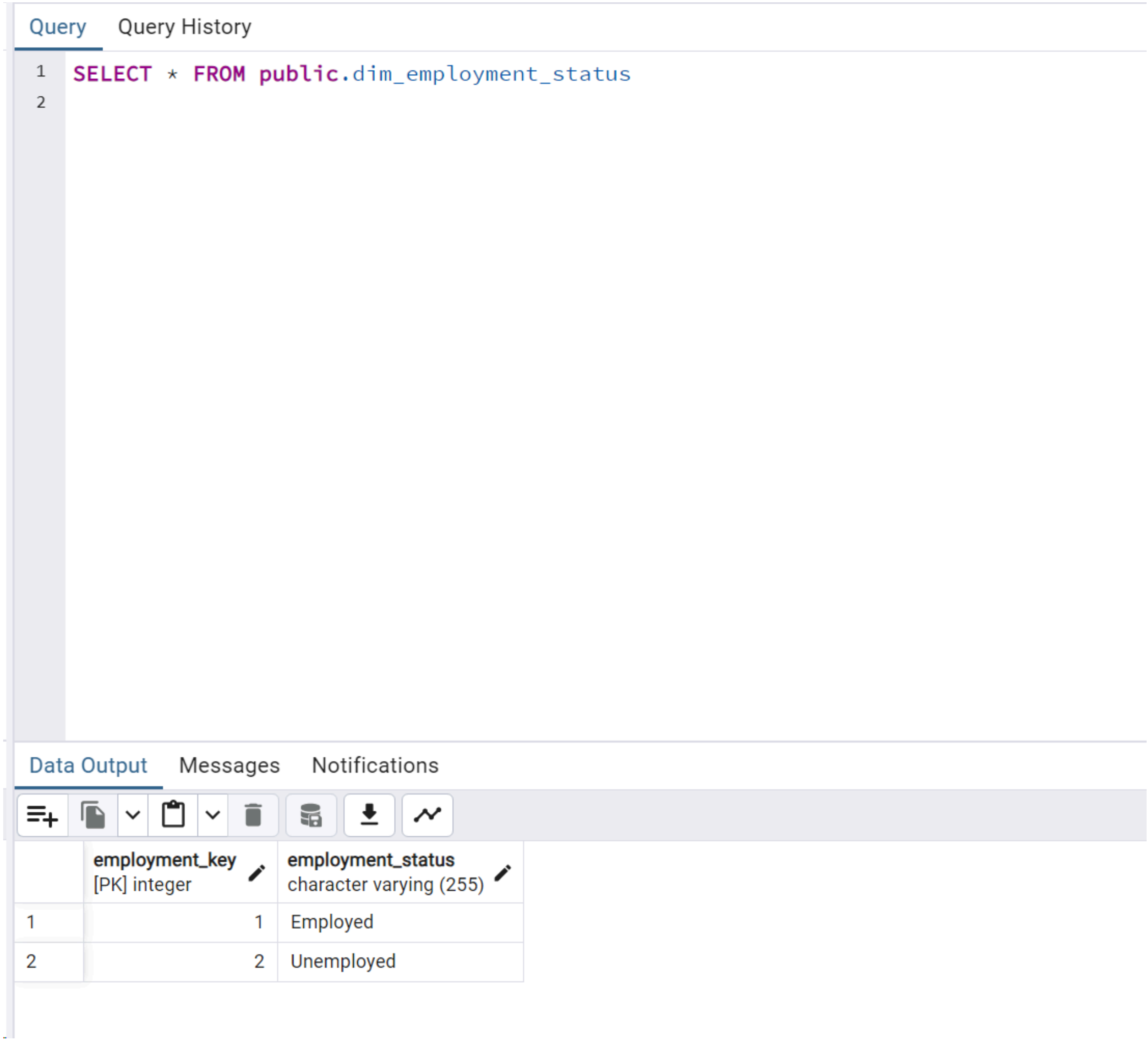
  Then the value in this row would be replaced with the average of the values in the rows where Labour Force Characteristics, NAICS, Gender, Age Group are the same.
  This however still resulted in some rows receiving an average of 0 due to all the rows matching the query also having values of 0.
- Binning values for the value attribute for each dataframe in order to reduce the complexity of the data, especially for GDP df (values > 1E10)
  - Screenshots of dimensions and fact table (sample).


Tables (6)
dim_date
dim_demographic
dim_education
dim_employment_status
dim_industry
fact_table



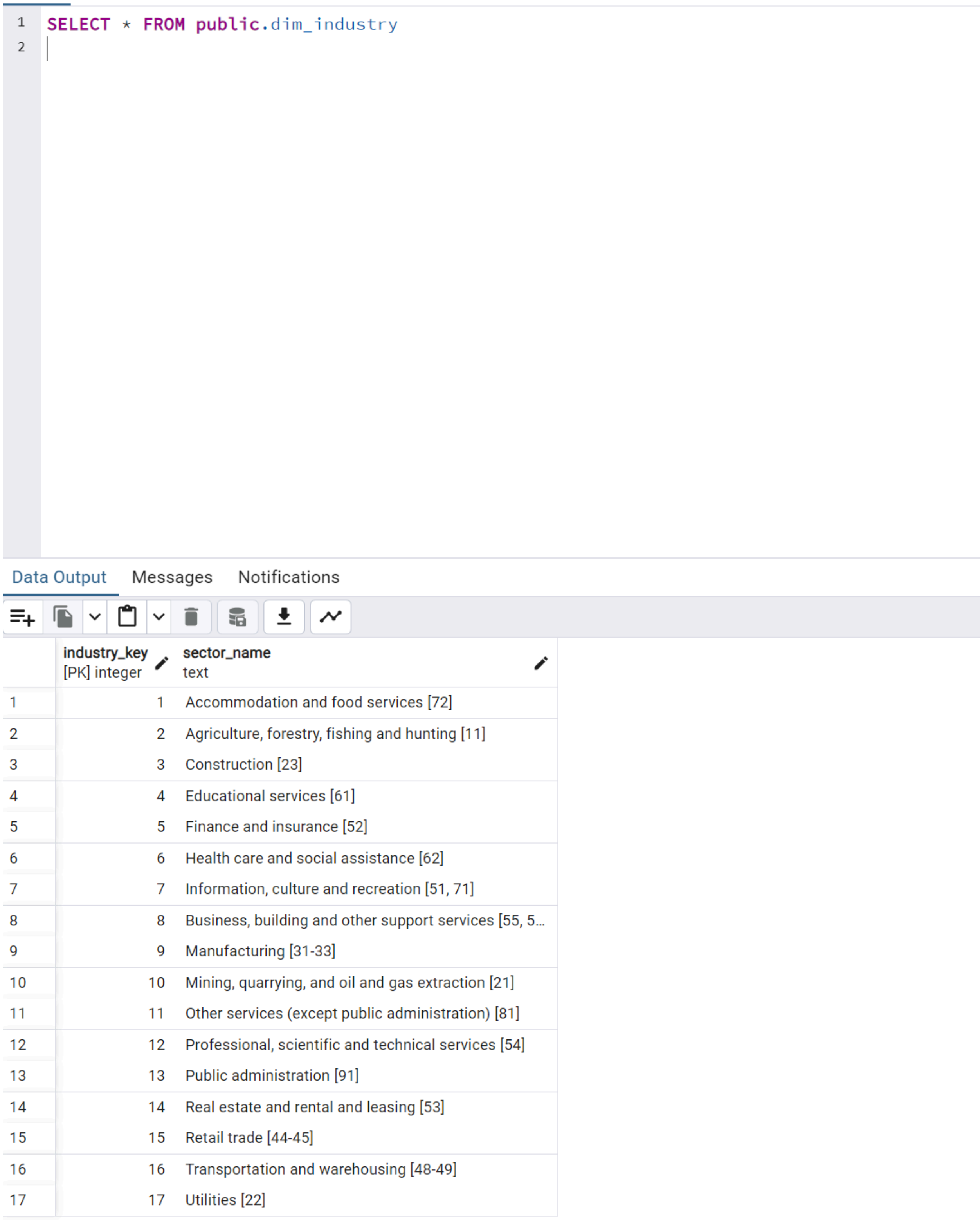
```sql
SELECT * FROM public.dim_education
```

	education_key [PK] integer	field_of_study text
1	1	Agriculture, natural resources and conservation [9]
2	2	Architecture, engineering and related technologies [8]
3	3	Business, management and public administration [5]
4	4	Education [1]
5	5	Health and related fields [10]
6	6	Humanities [3]
7	7	Mathematics, computer and information sciences [7]
8	8	Other [12]
9	9	Personal improvement and leisure [0]
10	10	Personal, protective and transportation services [11]
11	11	Physical and life sciences and technologies [6]
12	12	Social and behavioural sciences and law [4]
13	13	Unclassified
14	14	Visual and performing arts, and communications technologies [...



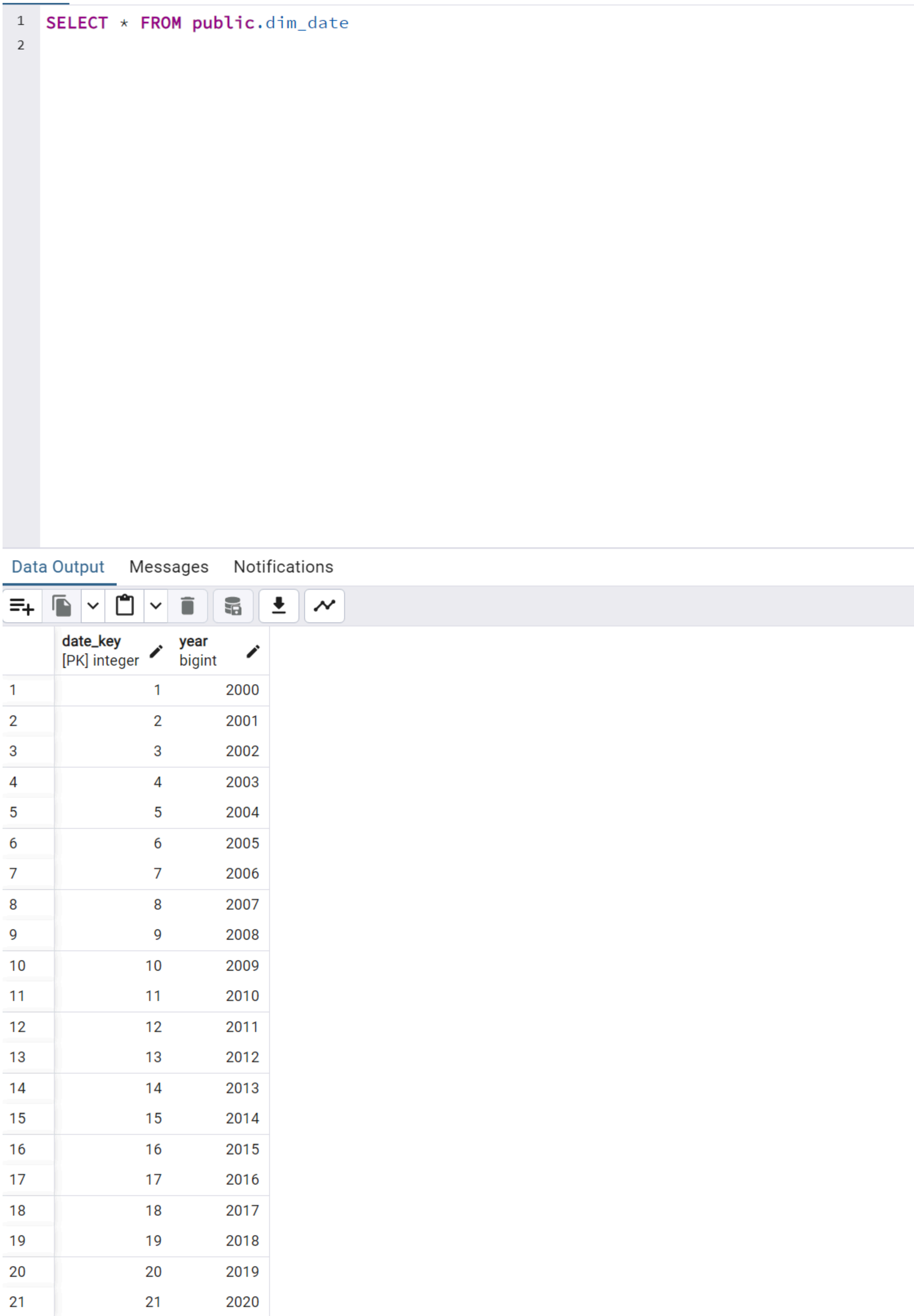
```sql
SELECT * FROM public.dim_employment_status

```

	employment_key [PK] integer	employment_status character varying (255)
1	1	Employed
2	2	Unemployed



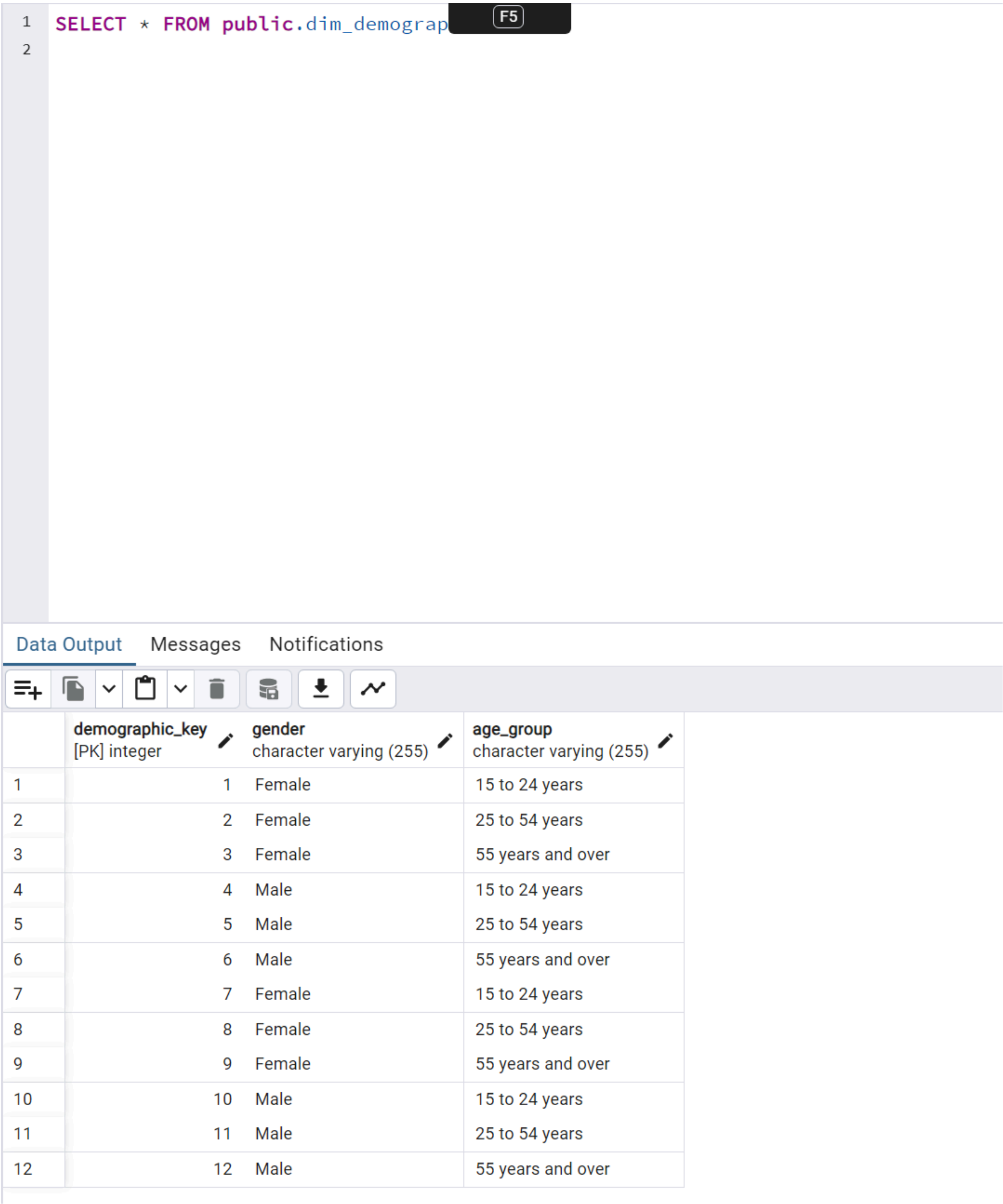
```sql
SELECT * FROM public.dim_industry
```

	industry_key [PK] integer	sector_name text
1	1	Accommodation and food services [72]
2	2	Agriculture, forestry, fishing and hunting [11]
3	3	Construction [23]
4	4	Educational services [61]
5	5	Finance and insurance [52]
6	6	Health care and social assistance [62]
7	7	Information, culture and recreation [51, 71]
8	8	Business, building and other support services [55, 5…
9	9	Manufacturing [31-33]
10	10	Mining, quarrying, and oil and gas extraction [21]
11	11	Other services (except public administration) [81]
12	12	Professional, scientific and technical services [54]
13	13	Public administration [91]
14	14	Real estate and rental and leasing [53]
15	15	Retail trade [44-45]
16	16	Transportation and warehousing [48-49]
17	17	Utilities [22]



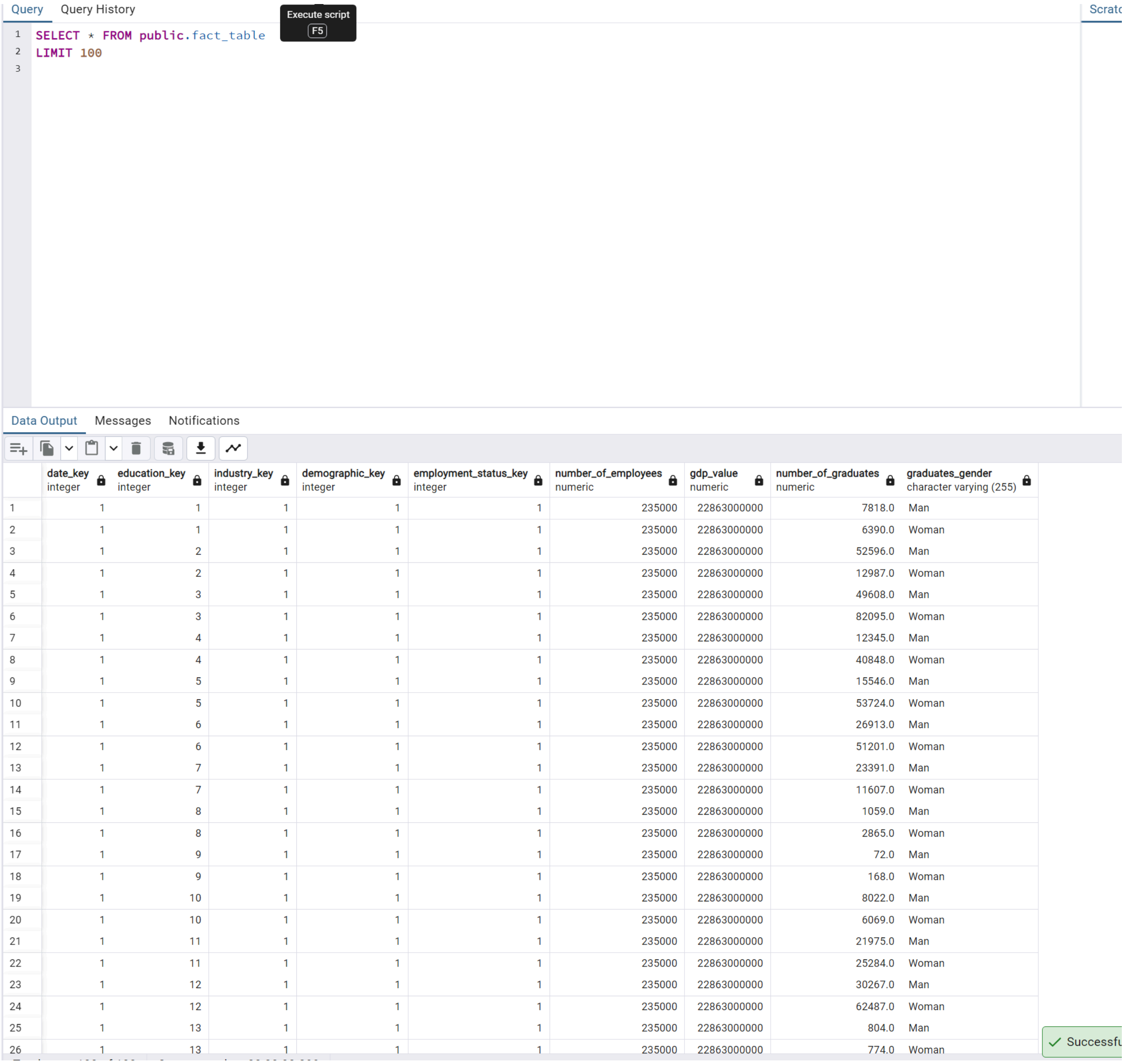
```sql
SELECT * FROM public.dim_date

```

	date_key [PK] integer	year bigint
1	1	2000
2	2	2001
3	3	2002
4	4	2003
5	5	2004
6	6	2005
7	7	2006
8	8	2007
9	9	2008
10	10	2009
11	11	2010
12	12	2011
13	13	2012
14	14	2013
15	15	2014
16	16	2015
17	17	2016
18	18	2017
19	19	2018
20	20	2019
21	21	2020



```sql
SELECT * FROM public.dim_demograp
```

	demographic_key [PK] integer	gender character varying (255)	age_group character varying (255)
1	1	Female	15 to 24 years
2	2	Female	25 to 54 years
3	3	Female	55 years and over
4	4	Male	15 to 24 years
5	5	Male	25 to 54 years
6	6	Male	55 years and over
7	7	Female	15 to 24 years
8	8	Female	25 to 54 years
9	9	Female	55 years and over
10	10	Male	15 to 24 years
11	11	Male	25 to 54 years
12	12	Male	55 years and over



```sql
SELECT * FROM public.fact_table
LIMIT 100
```

	date_key integer	education_key integer	industry_key integer	demographic_key integer	employment_status_key integer	number_of_employees numeric	gdp_value numeric	number_of_graduates numeric	graduates_gender character varying (255)
1	1	1	1	1	1	235000	22863000000	7818.0	Man
2	1	1	1	1	1	235000	22863000000	6390.0	Woman
3	1	2	1	1	1	235000	22863000000	52596.0	Man
4	1	2	1	1	1	235000	22863000000	12987.0	Woman
5	1	3	1	1	1	235000	22863000000	49608.0	Man
6	1	3	1	1	1	235000	22863000000	82095.0	Woman
7	1	4	1	1	1	235000	22863000000	12345.0	Man
8	1	4	1	1	1	235000	22863000000	40848.0	Woman
9	1	5	1	1	1	235000	22863000000	15546.0	Man
10	1	5	1	1	1	235000	22863000000	53724.0	Woman
11	1	6	1	1	1	235000	22863000000	26913.0	Man
12	1	6	1	1	1	235000	22863000000	51201.0	Woman
13	1	7	1	1	1	235000	22863000000	23391.0	Man
14	1	7	1	1	1	235000	22863000000	11607.0	Woman
15	1	8	1	1	1	235000	22863000000	1059.0	Man
16	1	8	1	1	1	235000	22863000000	2865.0	Woman
17	1	9	1	1	1	235000	22863000000	72.0	Man
18	1	9	1	1	1	235000	22863000000	168.0	Woman
19	1	10	1	1	1	235000	22863000000	8022.0	Man
20	1	10	1	1	1	235000	22863000000	6069.0	Woman
21	1	11	1	1	1	235000	22863000000	21975.0	Man
22	1	11	1	1	1	235000	22863000000	25284.0	Woman
23	1	12	1	1	1	235000	22863000000	30267.0	Man
24	1	12	1	1	1	235000	22863000000	62487.0	Woman
25	1	13	1	1	1	235000	22863000000	804.0	Man
26	1	13	1	1	1	235000	22863000000	774.0	Woman