ETL PROCESS

Analysis

Business processes identify:

- Job Applications
- Job Publications

Identified Metrics:

  - Company
  - Job
  - Date
  - Status
  - Location

For this case I'm not going to use a Slowly Changing Dimensions approach to keep track of the dimension history over the time, since it is a simple structure coming from the file, I will manage the changes like a different dimension value and all the values will be active all the time.

I let the benefits and skill out of this analysis since it is not required for the analysis capabilities.

Notes

1. Missing information: bucket name for S3, Redshift information like the server URL or host, database and schema.
2. I tried to access Raw as a s3 bucket, but it seems that it doesn't exist, or I don't have permission with the provided S3 keys.
3. I'm going to use the 300 jobs sample data set that I received.

Data profiling results

For this result, I used the data-sample.json file, since I couldn't access to the S3 location to get the files.

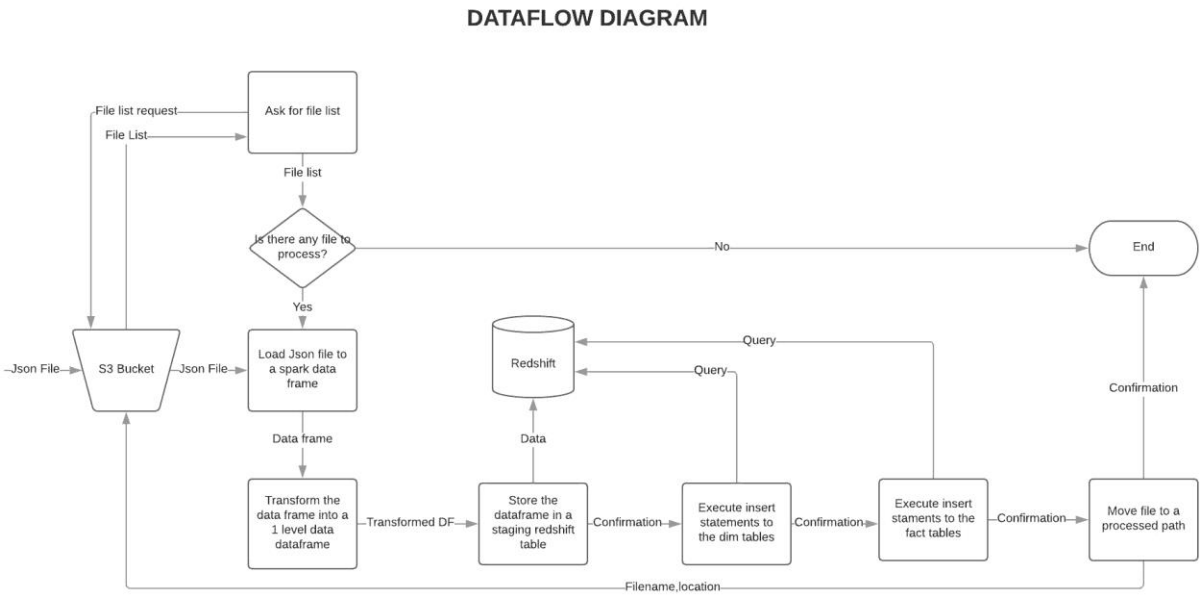
All the results are in the result.txt file in the same git directory.

Here are some results:

```
3) Applications analysis by company Sector
+--------------------+------------------+
|              sector|total_Applications|
+--------------------+------------------+
|            Services|               166|
|  Product Management|               192|
|               Sales|               167|
|         Engineering|               100|
|          Accounting|               169|
|Business Development|               205|
|Research and Deve...|               155|
|               Legal|               196|
|            Training|               153|
|           Marketing|               112|
|             Support|               182|
|     Human Resources|               193|
+--------------------+------------------+

4) Analysis of applications distributions based on age ranges
+---------+------------------+
|age_range|total_Applications|
+---------+------------------+
|      >70|               321|
|    11-20|              1829|
+---------+------------------+
```

Dataflow



When the script stars, the fist thing that is going to do is to check if there are available files to be processed on the S3 bucket and path like /raw/jobs. If there are multiples files, it will do the same following steps for each file.

Once a file is detected, then the process will read the json file and load it into a data frame.

Then the json file with the different levels is converted in to 1 level, like if it is a plain table for the database. With each field treated like a column.

After transforming the json file into a table structure, it is loaded to a dump table in redshift.

With the dump table in redshift, a series of queries are executed to take and insert the different values for the dimensions tables.

Once the dimensions values are available in the tables, the final queries will fill the fact tables with the information in the dump table and tie the keys with the dim tables.

After everything was process, then the file is moved from the original location to another path that indicates that the json file is already process.

The information is already in the fact and dimension tables, ready to start the data analysis.

Data Dictonary

Dim_Company

Field Name	Data Typ	Field Size	Description
Dim_company_id	bigint		Primary key of the table
Company_name	varchar(255)	255	The company name
Company_sector	varchar(255)	255	The sector that the company belongs
Insert_date	datetime		Date when the dimension value was inserted

Dim_Job

Field Name	Data Type	Field Size	Description
Dim_job_id	bigint		Primary key of the table
Id	varchar(36)	36	The Id Job from the file
Title	Varchar(255)	255	It stores the title of the advert job, even in the sample file the maximum character is 36, I set up to 255 just in case a long Title
Insert_date	datetime		Date when the dimension value was inserted

Dim_Date

Field Name	Data Type	Field Size	Description
Dim_date_id	bigint		Primary key of the table
Fulldate	date		The Full date in format mm/dd/yyyy
Day_of_week	Int		Stores the day number in the week
Day_of_month	Int		Stores the day number in the month
Day_of_year	Int		Stores the day number in the year
Week_of_year	Int		The week number in the year
Month	Int		The month number
Year	Int		The year of the date
Insert_date	datetime		Date when the dimension value was inserted

Dim_Status

Field Name	Data Type	Field Size	Description
Dim_status_id	bigint		Primary key of the table
Status	Varchar(8)	8	The possible Status are Active,Deleted, Inactive
Insert_date	datetime		Date when the dimension value was inserted

Dim_Location

Field Name	Data Type	Field Size	Description
Dim_location_id	bigint		Primary key of the table
City	Varchar(255)	255	Stores the job city, even the maximum length in the files is 20, I will leave this on 255 just in case a bigger value comes
Insert_date	datetime		Date when the dimension value was inserted

Dim_Advert

Field Name	Data Type	Field Size	Description
Dim_advert_id	bigint		Primary key of the table
Id	varchar(36)		The Id Job from the file
Apply_url	Varchar(2500)	2500	The URL to apply to the job, even the maximum is like 1375, I let a bigger space in case is needed
Insert_date	datetime		Date when the dimension value was inserted

Dim_Applicant

Field Name	Data Type	Field Size	Description
Dim_applicant_id	bigint		Primary key of the table
First_name	Varchar(255)		Name of the applicant
Last_name	Varchar(255)		Last Name of the applicant
Insert_date	datetime		Date when the dimension value was inserted

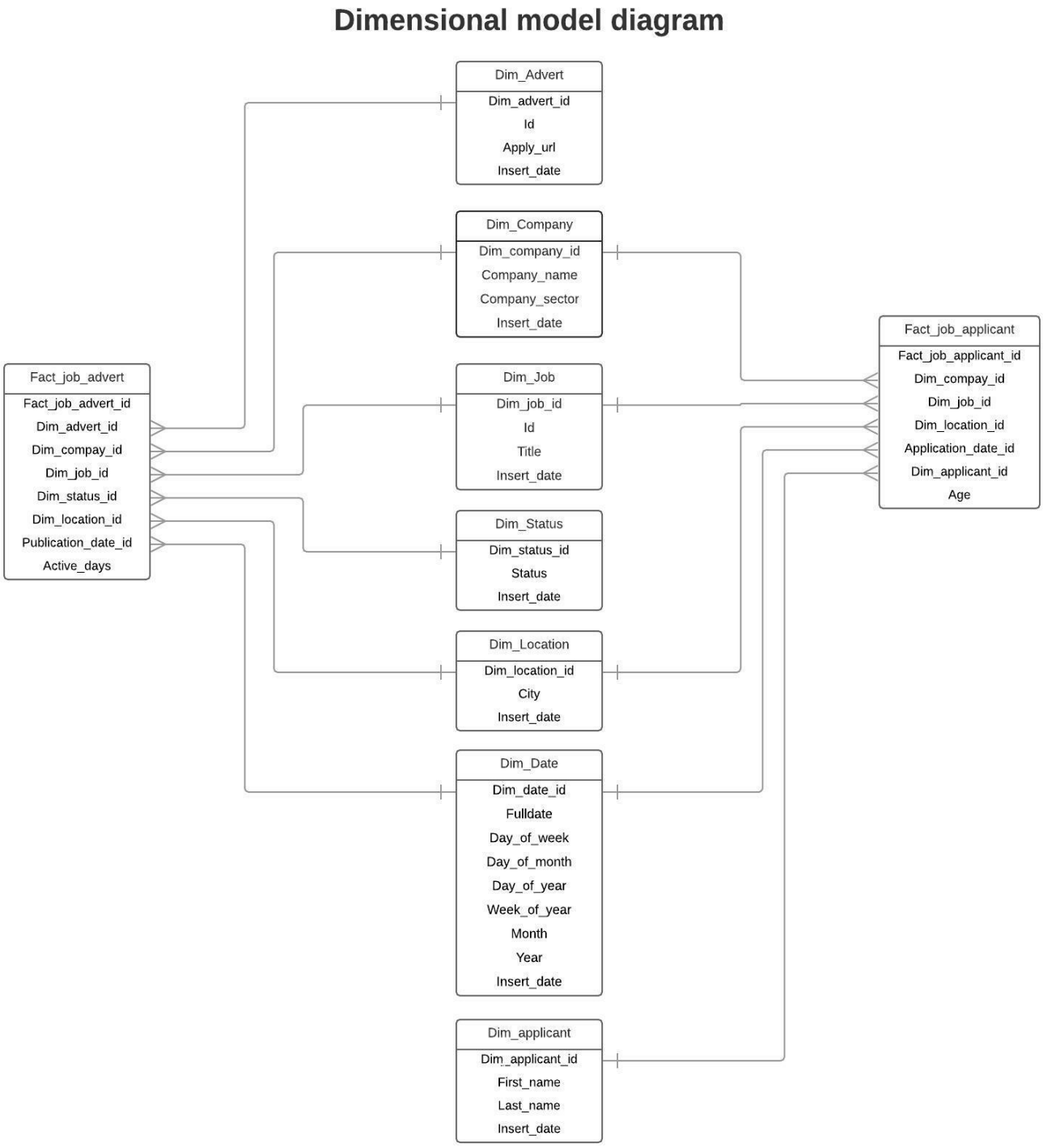
Fact_job_advert

Field Name	Data Type	Field Size	Description
Fact_job_advert_id	bigint		Primary key of the table
Dim_advert_id	bigint		Foreign key to the advert dimension
Dim_compay_id	bigint		Foreign key to the company dimension
Dim_job_id	bigint		Foreign key to the job dimension
Dim_status_id	bigint		Foreign key to the status dimension
Dim_location_id	bigint		Foreign key to the location dimension
Publication_date_id	bigint		Foreign key to the date dimension
Active_days	int		Date when the dimension value was inserted

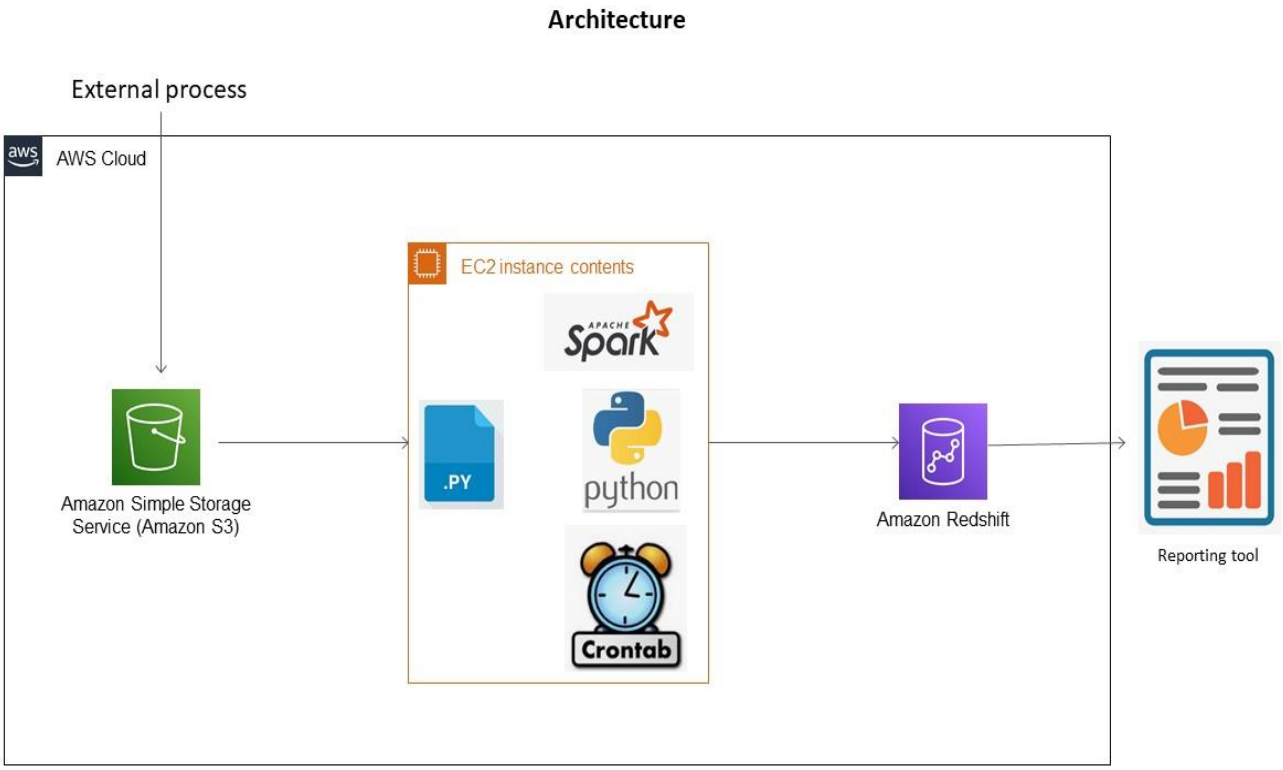
Fact_job_applicant

Field Name	Data Type	Field Size	Description
Fact_job_applicant_id	bigint		Primary key of the table
Dim_company_id	bigint		Foreign key to the company dimension
Dim_job_id	bigint		Foreign key to the job dimension
Dim_location_id	bigint		Foreign key to the location dimension
Application_date_id	bigint		Foreign key to the date dimension
Dim_applicant_id	bigint		Foreign key to the applicant dimension
Age	Int		Date when the dimension value was inserted

Dimensional model diagram



Architecture



The pipeline can be implemented in different services like AWS glue, Amazon EMR, or can be setup the environment in a EC2 instance. For this case, I built a general python script that can be ran in a EC2.

In our EC2 instance Apache Spark and python is needed and copy the python script and the .ini file.

A crontab is needed to schedule when the pipeline should run.

A bucket in S3 will serve as the file location, and inside that bucket 2 paths needs to be configured, one will be to receive the json files that will be processed, and the other path will have the processed files.

And redshift should be configured, and have the tables contained in the redshiftsript.sql.

Also Jobs_ETL.log is configured to track error, it should be in the same location as the python script.

In the .ini file will be the configurations like s3 credentials, bucket, path, and redshift configuration.

For the .ini file here are some notes:

- The paths configuration should not start and end in /. Just the ones that are in the middle to difference the sublocations.
- The dump_table configuration should include the shema in the format: Schema.table
- All the text configuration should not be inside “ ” or ‘ ’

Notes:

1. Make sure that redshift.jdbc42 driver is installed and available to be used by the spark context.
2. Make sure that the hadoop-aws package is available to be used by spark.
3. Pypasrk and psycopg2 libraries available as well.