

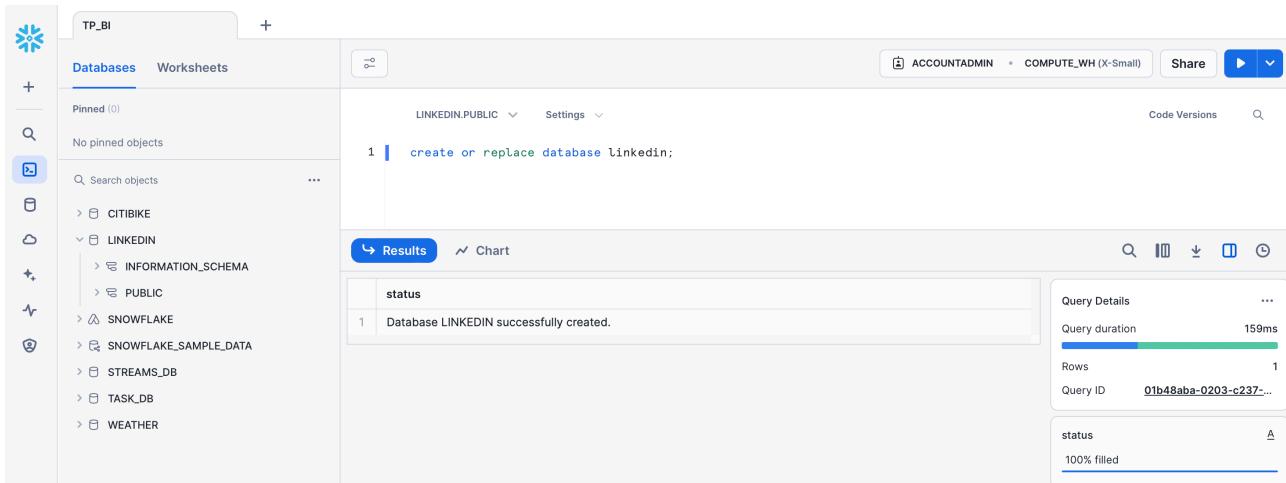
Projet module architecture BI

Les fichiers sont dans un bucket S3 public s3://snowflake-lab-bucket/.

Chargement des données

Voici la liste des étapes pour charger les données dans une base de données Snowflake

- Créez une nouvelle base de données « linkedin »



The screenshot shows the Snowflake interface with the database 'TP_BI' selected. In the left sidebar, under 'Databases', there is a tree view with nodes like CITIBIKE, LINKEDIN, SNOWFLAKE, SNOWFLAKE_SAMPLE_DATA, STREAMS_DB, TASK_DB, and WEATHER. The 'LINKEDIN' node is expanded, showing its sub-nodes: INFORMATION_SCHEMA and PUBLIC. In the main query editor area, the database is set to 'LINKEDIN.PUBLIC'. A query is being run:
`create or replace database linkedin;`
The results pane shows the status: 'Database LINKEDIN successfully created.' The right panel displays 'Query Details' with a duration of 159ms and 1 row processed, and a progress bar indicating 100% filled.

- Créez un stage qui spécifie l'emplacement du bucket S3.



The screenshot shows the Snowflake interface with the database 'TP_BI' selected. In the left sidebar, under 'Databases', the 'LINKEDIN' node is expanded, showing its sub-nodes: INFORMATION_SCHEMA and PUBLIC. The 'PUBLIC' node is also expanded, showing its 'Stages' sub-node. A new stage named 'STAGE_LINKEDIN' is listed under 'Stages'. In the main query editor area, the database is set to 'LINKEDIN.PUBLIC'. A query is being run:
`create or replace database linkedin;`
`create stage stage_linkedin url = 's3://snowflake-lab-bucket/' ;`
The results pane is currently empty. The right panel displays 'Query Details' with a duration of 20ms.

Vérification du contenu de notre stage.

```

TP_BI
+ Databases Worksheets
Pinned (0)
No pinned objects
Q Search objects ...
CITIBIKE
LINKEDIN
  INFORMATION_SCHEMA
    PUBLIC
      Stages
        STAGE_LINKEDIN
SNOWFLAKE
SNOWFLAKE_SAMPLE_DATA
STREAMS_DB
TASK_DB
WEATHER

```

LINKEDIN.PUBLIC Settings Code Versions

```

1 create or replace database linkedin;
2
3 create stage stage_linkedin url = 's3://snowflake-lab-bucket/' ;
4
5 list @stage_linkedin ;
6
7
8
9
10
11
12
13
14
15
16
17
18
19

```

Results Chart

name	size	md5	last_modified
s3://snowflake-lab-bucket/benefits.json	2423811	18534fc12a233134736d0fd892f336be	Wed, 22 May 2024 09:14:04 GMT
s3://snowflake-lab-bucket/companies.csv	10953429	0e10a6f6d4327decb26cae0df5cb3	Wed, 22 May 2024 09:13:20 GMT
s3://snowflake-lab-bucket/company_industries.json	957316	ea6641581054bad7b7d9b69624082e0	Wed, 22 May 2024 09:17:14 GMT
s3://snowflake-lab-bucket/company_specialties.csv	2036857	fd990cb73b7iae715c1144cb3b27b1a8	Wed, 22 May 2024 09:13:20 GMT
s3://snowflake-lab-bucket/employee_counts.json	1787096	036320e0369c820849e7a5dba9c6dda	Wed, 22 May 2024 09:17:15 GMT
s3://snowflake-lab-bucket/industries.json	17927	95569a39f22d9d1b255ce647495f47ea	Wed, 22 May 2024 09:17:15 GMT
s3://snowflake-lab-bucket/job_industries.json	2518017	2841f3557069a9bcaa8f5325de38bdca	Wed, 22 May 2024 09:17:16 GMT
s3://snowflake-lab-bucket/job_postings.csv	134845243	a1878ad1be787295e6f6997c39649d24-9	Wed, 22 May 2024 09:13:20 GMT
s3://snowflake-lab-bucket/job_skills.json	3302621	100f7a4f46fa4e17409995c5f21f46bc	Wed, 22 May 2024 09:14:05 GMT
s3://snowflake-lab-bucket/salaries.json	2953485	8178c9d37052a24e2848594226f9798a	Wed, 22 May 2024 09:14:06 GMT
s3://snowflake-lab-bucket/skills.json	2340	b95760af09153a9c96fc852ef382101	Wed, 22 May 2024 09:14:07 GMT

- Créez les files formats qui correspondent à la structure de données à charger

app.snowflake.com TP_BI - Snowflake Modern-Data-Ware...

```

TP_BI
+ Databases Worksheets
Pinned (0)
No pinned objects
Q Search objects ...
CITIBIKE
LINKEDIN
  INFORMATION_SCHEMA
    PUBLIC
      Tables
        BENEFITS
        COMPANIES
        COMPANY_INDUSTRIES
        COMPANY_SPECIALITIES
        EMPLOYEE_COUNTS
        INDUSTRIES
        JOBS_POSTING
        JOB_INDUSTRIES
        JOB_SKILLS
        SALARIES
        SKILLS
      Stages
        STAGE_LINKEDIN
      File Formats

```

LINKEDIN.PUBLIC Settings Code Versions

```

1 create or replace database linkedin;
2
3 create stage stage_linkedin url = 's3://snowflake-lab-bucket/' ;
4
5 list @stage_linkedin ;
6
7
8
9
10
11
12
13
14
15
16
17
18
19

```

Results Chart

status
File format CSV successfully created.

Query Details

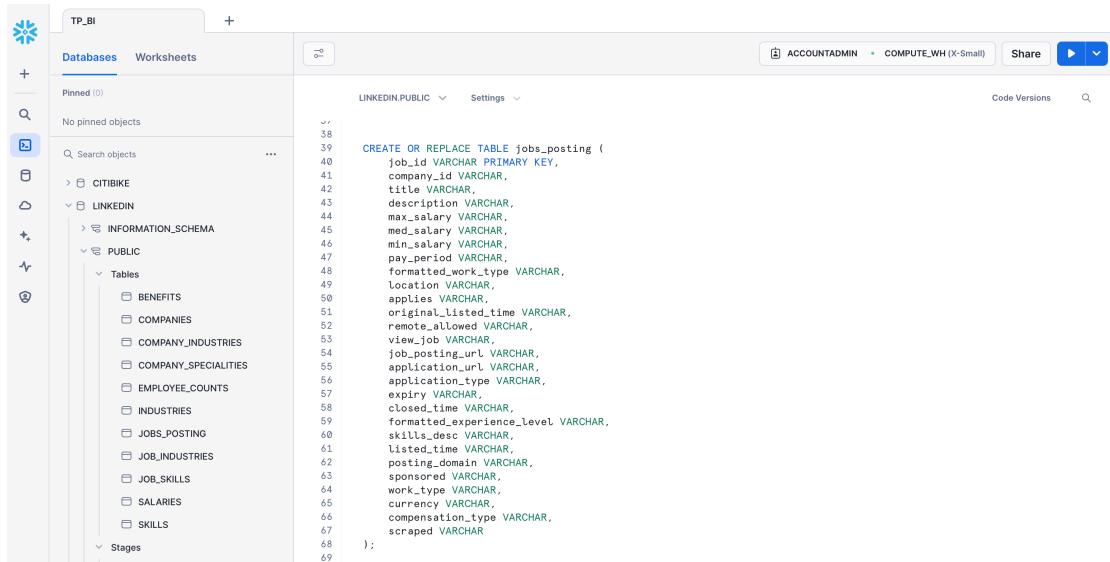
- Query duration: 60ms
- Rows: 1
- Query ID: 01b48b00-0203-c240-...

status

- 100% filled

- Créez les différentes tables de données.

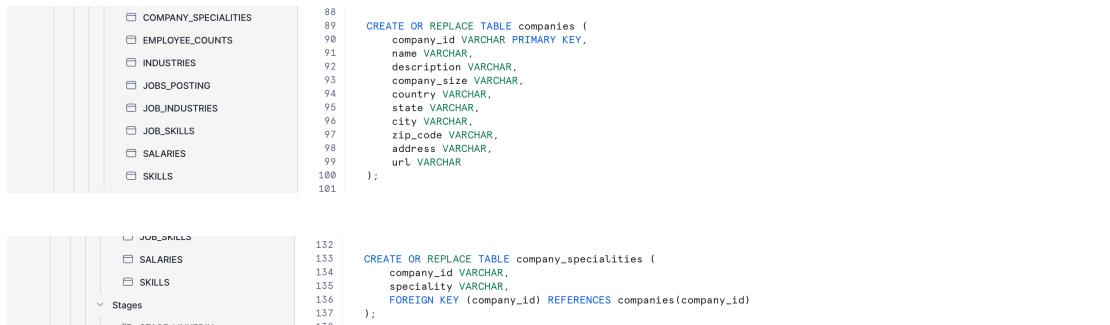
Dans un premier temps, nous créons toutes les tables demandées en fixant le type de chaque colonne à VARCHAR afin de savoir le type de chaque colonne. Ensuite, nous chargeons toutes les données dans chacune des tables, et une fois les données chargées, nous affinerons le type de chaque colonne de chaque table.



```

CREATE OR REPLACE TABLE jobs_posting (
    job_id VARCHAR PRIMARY KEY,
    company_id VARCHAR,
    title VARCHAR,
    description VARCHAR,
    max_salary VARCHAR,
    med_salary VARCHAR,
    min_salary VARCHAR,
    pay_period VARCHAR,
    formatted_pay_type VARCHAR,
    location VARCHAR,
    applies VARCHAR,
    original_listed_time VARCHAR,
    remote_allowed VARCHAR,
    view_job VARCHAR,
    job_posting_url VARCHAR,
    application_url VARCHAR,
    application_type VARCHAR,
    expiry VARCHAR,
    closed_time VARCHAR,
    formatted_experience_level VARCHAR,
    skills_desc VARCHAR,
    listed_time VARCHAR,
    posting_domain VARCHAR,
    sponsored VARCHAR,
    work_type VARCHAR,
    currency VARCHAR,
    compensation_type VARCHAR,
    scraped VARCHAR
);

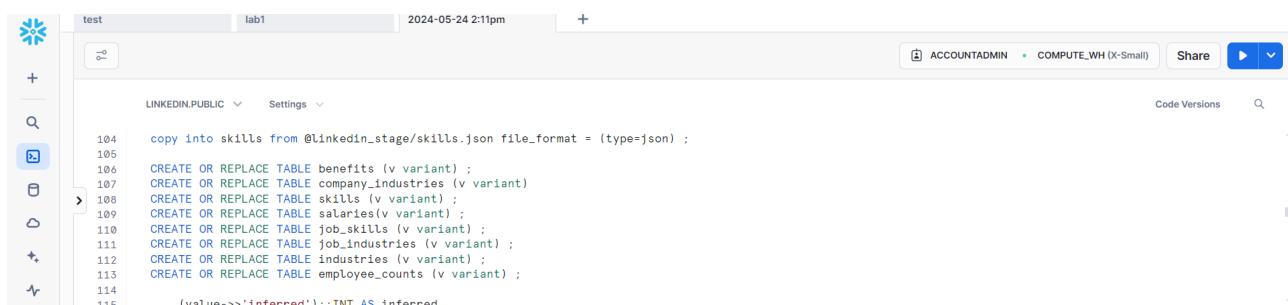
```



```

CREATE OR REPLACE TABLE companies (
    company_id VARCHAR PRIMARY KEY,
    name VARCHAR,
    description VARCHAR,
    company_size VARCHAR,
    country VARCHAR,
    state VARCHAR,
    city VARCHAR,
    zip_code VARCHAR,
    address VARCHAR,
    url VARCHAR
);

```



```

CREATE OR REPLACE TABLE company_specialties (
    company_id VARCHAR,
    speciality VARCHAR,
    FOREIGN KEY (company_id) REFERENCES companies(company_id)
);

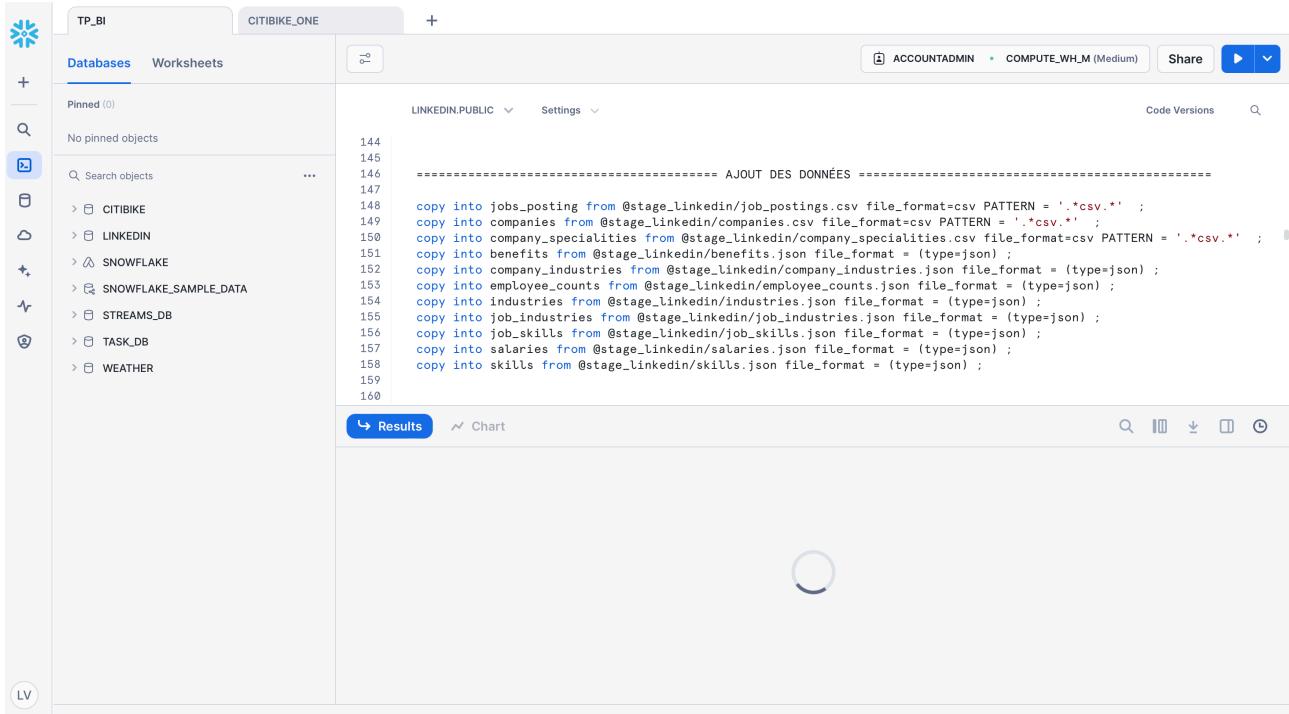
```

```

copy into skills from @linkedin_stage/skills.json file_format = (type=json) ;
CREATE OR REPLACE TABLE benefits (v variant) ;
CREATE OR REPLACE TABLE company_industries (v variant) ;
CREATE OR REPLACE TABLE skills (v variant) ;
CREATE OR REPLACE TABLE salaries(v variant) ;
CREATE OR REPLACE TABLE job_skills (v variant) ;
CREATE OR REPLACE TABLE job_industries (v variant) ;
CREATE OR REPLACE TABLE industries (v variant) ;
CREATE OR REPLACE TABLE employee_counts (v variant) ;
value->'inferred')--TNT AS inferred

```

Chargement des données dans les tables.

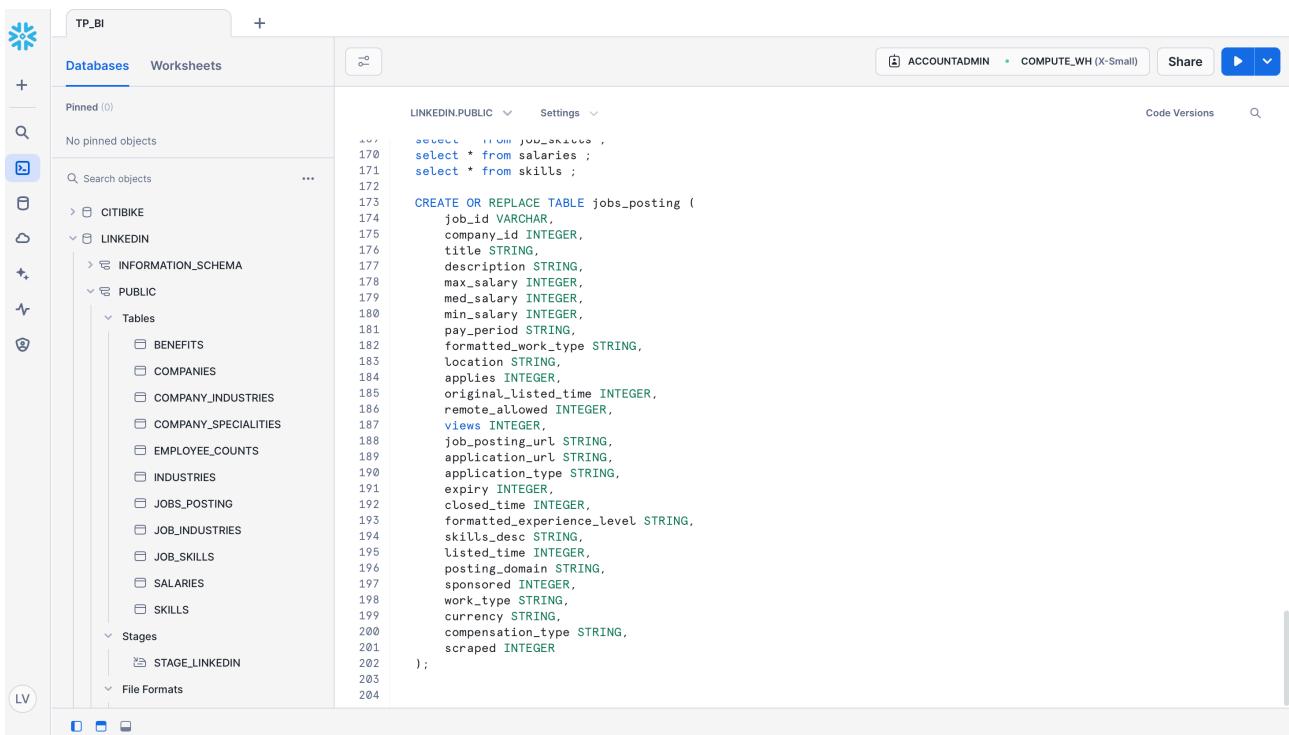


```

TP_BI CITIBIKE_ONE +
Databases Worksheets
Pinned (0)
No pinned objects
Q. Search objects ...
> CITIBIKE
> LINKEDIN
> SNOWFLAKE
> SNOWFLAKE_SAMPLE_DATA
> STREAMS_DB
> TASK_DB
> WEATHER
LINKEDIN.PUBLIC Settings
144 ===== AJOUT DES DONNÉES =====
145
146 copy into jobs_posting from @stage_linkedin/job_postings.csv file_format=csv PATTERN = '.*csv.*' ;
147
148 copy into companies from @stage_linkedin/companies.csv file_format=csv PATTERN = '.*csv.*' ;
149 copy into company_specialties from @stage_linkedin/company_specialties.csv file_format=csv PATTERN = '.*csv.*' ;
150
151 copy into benefits from @stage_linkedin/benefits.json file_format = (type=json) ;
152 copy into company_industries from @stage_linkedin/company_industries.json file_format = (type=json) ;
153 copy into employee_counts from @stage_linkedin/employee_counts.json file_format = (type=json) ;
154 copy into industries from @stage_linkedin/industries.json file_format = (type=json) ;
155 copy into job_industries from @stage_linkedin/job_industries.json file_format = (type=json) ;
156 copy into job_skills from @stage_linkedin/job_skills.json file_format = (type=json) ;
157 copy into salaries from @stage_linkedin/salaries.json file_format = (type=json) ;
158 copy into skills from @stage_linkedin/skills.json file_format = (type=json) ;
159
160
Results Chart

```

Modification des types de chaque colonne de chacune des tables dont les données proviennent d'un fichier csv. Nous chargerons les données, provenant d'un fichier json, dans les tables après avoir effectué une view sur chacune d'elles prochainement.



```

TP_BI +
Databases Worksheets
Pinned (0)
No pinned objects
Q. Search objects ...
> CITIBIKE
> LINKEDIN
> INFORMATION_SCHEMA
> PUBLIC
Tables
  BENEFITS
  COMPANIES
  COMPANY_INDUSTRIES
  COMPANY_SPECIALITIES
  EMPLOYEE_COUNTS
  INDUSTRIES
  JOBS_POSTING
  JOB_INDUSTRIES
  JOB_SKILLS
  SALARIES
  SKILLS
Stages
  STAGE_LINKEDIN
File Formats
LINKEDIN.PUBLIC Settings
173 select * from jobs_posting ;
174 select * from salaries ;
175 select * from skills ;
176
177 CREATE OR REPLACE TABLE jobs_posting (
178   job_id VARCHAR,
179   company_id INTEGER,
180   title STRING,
181   description STRING,
182   max_salary INTEGER,
183   med_salary INTEGER,
184   min_salary INTEGER,
185   pay_period STRING,
186   formatted_work_type STRING,
187   location STRING,
188   applies INTEGER,
189   original_listed_time INTEGER,
190   remote_allowed INTEGER,
191   views INTEGER,
192   job_posting_url STRING,
193   application_url STRING,
194   application_type STRING,
195   expiry INTEGER,
196   closed_time INTEGER,
197   formatted_experience_level STRING,
198   skills_desc STRING,
199   listed_time INTEGER,
200   posting_domain STRING,
201   sponsored INTEGER,
202   work_type STRING,
203   currency STRING,
204   compensation_type STRING,
205   scraped INTEGER
206 );

```

```

    200     compensation_type STRING,
    201     scraped INTEGER
    202 );
    203
    204 CREATE OR REPLACE TABLE companies (
    205     company_id INTEGER PRIMARY KEY,
    206     name VARCHAR,
    207     description VARCHAR,
    208     company_size INTEGER,
    209     country VARCHAR,
    210     state VARCHAR,
    211     city VARCHAR,
    212     zip_code VARCHAR,
    213     address VARCHAR,
    214     url VARCHAR
    215 );
    216
    217 CREATE OR REPLACE TABLE company_specialties (
    218     company_id INT,
    219     speciality VARCHAR,
    220     FOREIGN KEY (company_id) REFERENCES companies(company_id)
    221 );
    222

```

Une fois les tables de nouveau créées avec les bons types, nous rechargeons toutes les données dans ces tables.

- Effectuez les transformations nécessaires pour rendre les données exploitables

Ici, nous allons créer pour chaque fichier JSON une View.

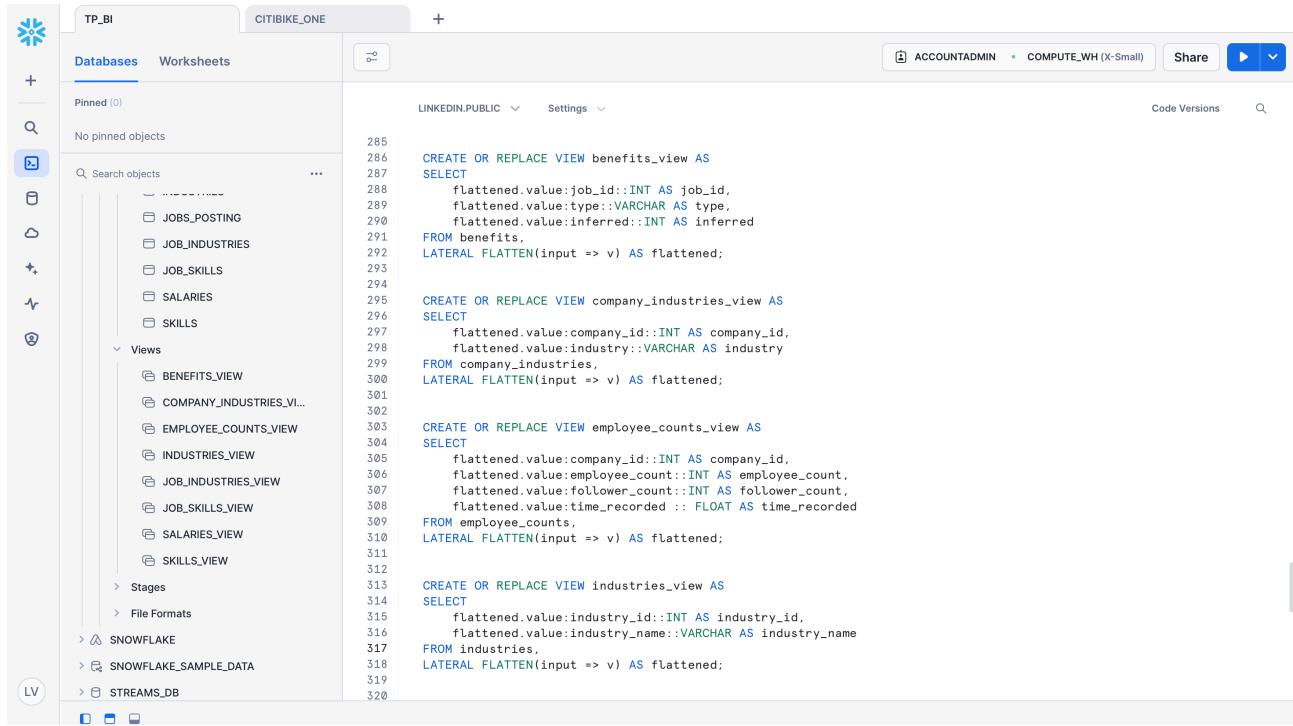
```

    283
    284 ====== CRÉATION DES VIEWS ======
    285
    286 CREATE OR REPLACE VIEW benefits_view AS
    287 SELECT
    288     v:job_id::INT AS job_id,
    289     v:type::VARCHAR AS type,
    290     v:inferred::INT as inferred
    291 FROM benefits;
    292
    293
    294 CREATE OR REPLACE VIEW company_industries_view AS
    295 SELECT
    296     v:company_id::INT AS company_id,
    297     v:industry::VARCHAR AS industry
    298 FROM company_industries;
    299
    300 SELECT * FROM company_industries_view
    301
    302
    303 CREATE OR REPLACE VIEW employee_counts_view AS
    304 SELECT
    305     v:company_id::INT AS company_id,
    306     v:employee_count::INT AS employee_count,
    307     v:follower_count::INT AS follower_count,
    308     to_timestamp(v:time_recorded) AS time_recorded
    309 FROM employee_counts;
    310
    311
    312 CREATE OR REPLACE VIEW industries_view AS
    313 SELECT
    314     v:industry_id::INT AS industry_id,
    315     v:industry_name::VARCHAR AS industry_name
    316 FROM industries;
    317
    318
    319 CREATE OR REPLACE VIEW job_industries_view AS
    320 SELECT
    321     v:job_id::INT AS job_id,
    322     v:industry_id::INT AS industry_id
    323 FROM job_industries;
    324
    325
    326 CREATE OR REPLACE VIEW job_skills_view AS
    327 SELECT
    328     v:job_id::INT AS job_id,
    329     v:skill_abr::VARCHAR AS skill_abr
    330 FROM job.skills;
    331
    332
    333 CREATE OR REPLACE VIEW salaries_view AS
    334 SELECT
    335     v:salary_id::INT AS salary_id,
    336     v:job_id::INT AS job_id,
    337     v:max_salary::INT AS max_salary,
    338     v:med_salary::INT AS med_salary,
    339     v:min_salary::INT AS min_salary,
    340     v:pay_period::VARCHAR AS pay_period,
    341     v:currency::VARCHAR AS currency,
    342     v:compensation_type::VARCHAR AS compensation_type
    343 FROM salaries;
    344
    345
    346 CREATE OR REPLACE VIEW skills_view AS
    347 SELECT
    348     v:skill_abr::VARCHAR AS skill_abr,
    349     v:skill_name::VARCHAR AS skill_name,
    350     FROM skills;
    351
    352
    353
    354
    355
    356
    357
    358
    359
    360
    361
    362
    363
    364
    365
    366
    367
    368
    369
    370
    371
    372
    373
    374
    375

```

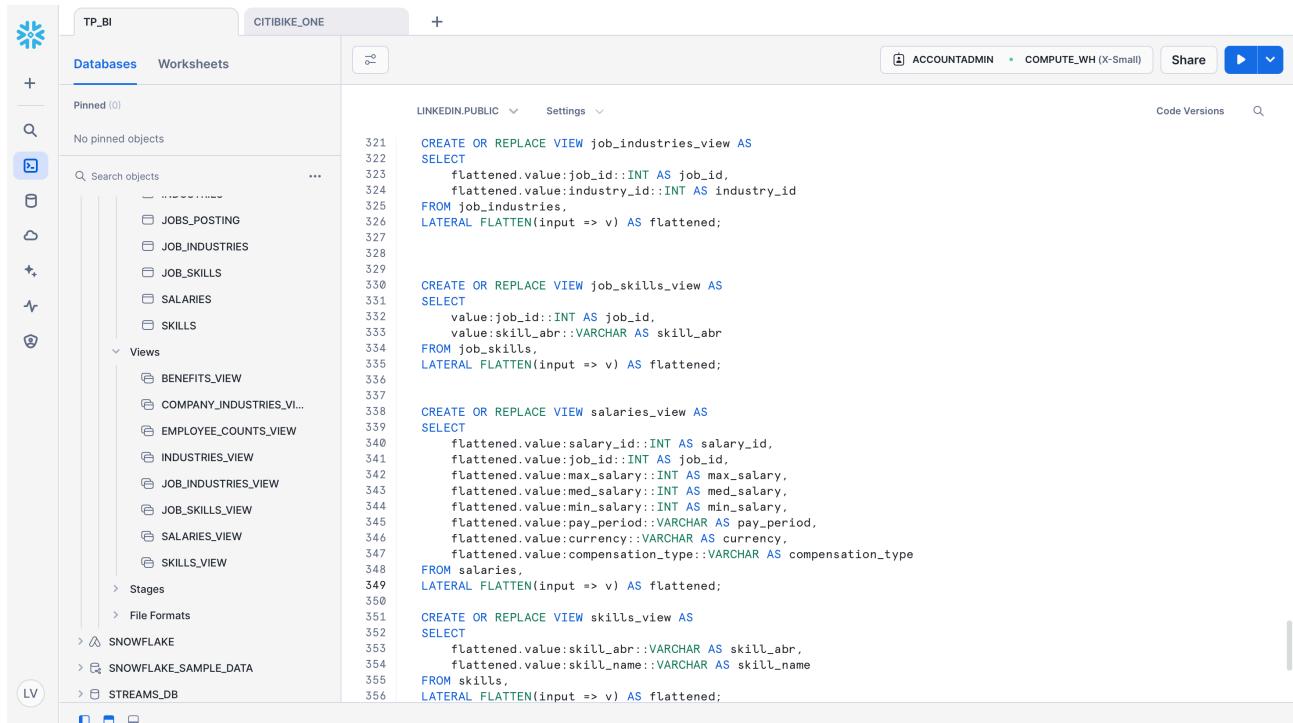
Après création des views, nous faisons face au problème que les views ne contiennent pas les données de la table associée.

Pour pallier le problème rencontré, nous changeons la syntaxe de la création de nos views comme ci-dessous :



```
TP BI CITIBIKE_ONE +
Databases Worksheets
Pinned (0)
No pinned objects
Search objects ...
JOBS_POSTING
JOB_INDUSTRIES
JOB_SKILLS
SALARIES
SKILLS
Views
BENEFITS_VIEW
COMPANY_INDUSTRIES_VI...
EMPLOYEE_COUNTS_VIEW
INDUSTRIES_VIEW
JOB_INDUSTRIES_VIEW
JOB_SKILLS_VIEW
SALARIES_VIEW
SKILLS_VIEW
Stages
File Formats
SNOWFLAKE
SNOWFLAKE_SAMPLE_DATA
STREAMS_DB
Code Versions Share
```

```
285
286 CREATE OR REPLACE VIEW benefits_view AS
287   SELECT
288     flattened.value:job_id::INT AS job_id,
289     flattened.value:type::VARCHAR AS type,
290     flattened.value:inferred::INT AS inferred
291   FROM benefits,
292     LATERAL FLATTEN(input => v) AS flattened;
293
294
295 CREATE OR REPLACE VIEW company_industries_view AS
296   SELECT
297     flattened.value:company_id::INT AS company_id,
298     flattened.value:industry::VARCHAR AS industry
299   FROM company_industries,
300     LATERAL FLATTEN(input => v) AS flattened;
301
302
303 CREATE OR REPLACE VIEW employee_counts_view AS
304   SELECT
305     flattened.value:company_id::INT AS company_id,
306     flattened.value:employee_count::INT AS employee_count,
307     flattened.value:follower_count::INT AS follower_count,
308     flattened.value:time_recorded :: FLOAT AS time_recorded
309   FROM employee_counts,
310     LATERAL FLATTEN(input => v) AS flattened;
311
312
313 CREATE OR REPLACE VIEW industries_view AS
314   SELECT
315     flattened.value:industry_id::INT AS industry_id,
316     flattened.value:industry_name::VARCHAR AS industry_name
317   FROM industries,
318     LATERAL FLATTEN(input => v) AS flattened;
319
320
```



```
TP BI CITIBIKE_ONE +
Databases Worksheets
Pinned (0)
No pinned objects
Search objects ...
JOBS_POSTING
JOB_INDUSTRIES
JOB_SKILLS
SALARIES
SKILLS
Views
BENEFITS_VIEW
COMPANY_INDUSTRIES_VI...
EMPLOYEE_COUNTS_VIEW
INDUSTRIES_VIEW
JOB_INDUSTRIES_VIEW
JOB_SKILLS_VIEW
SALARIES_VIEW
SKILLS_VIEW
Stages
File Formats
SNOWFLAKE
SNOWFLAKE_SAMPLE_DATA
STREAMS_DB
Code Versions Share
```

```
321
322 CREATE OR REPLACE VIEW job_industries_view AS
323   SELECT
324     flattened.value:job_id::INT AS job_id,
325     flattened.value:industry_id::INT AS industry_id
326   FROM job_industries,
327     LATERAL FLATTEN(input => v) AS flattened;
328
329
330 CREATE OR REPLACE VIEW job_skills_view AS
331   SELECT
332     value:job_id::INT AS job_id,
333     value:skill_abr::VARCHAR AS skill_abr
334   FROM job_skills,
335     LATERAL FLATTEN(input => v) AS flattened;
336
337
338 CREATE OR REPLACE VIEW salaries_view AS
339   SELECT
340     flattened.value:salary_id::INT AS salary_id,
341     flattened.value:job_id::INT AS job_id,
342     flattened.value:max_salary::INT AS max_salary,
343     flattened.value:med_salary::INT AS med_salary,
344     flattened.value:min_salary::INT AS min_salary,
345     flattened.value:pay_period::VARCHAR AS pay_period,
346     flattened.value:currencty::VARCHAR AS currency,
347     flattened.value:compensation_type::VARCHAR AS compensation_type
348   FROM salaries,
349     LATERAL FLATTEN(input => v) AS flattened;
350
351 CREATE OR REPLACE VIEW skills_view AS
352   SELECT
353     flattened.value:skill_abr::VARCHAR AS skill_abr,
354     flattened.value:skill_name::VARCHAR AS skill_name
355   FROM skills,
356     LATERAL FLATTEN(input => v) AS flattened;
```

Analyse des données:

Dans un premier temps, nous faisons le choix de vérifier nos requêtes SQL sur Snowflake. Une fois le résultat correct, nous ferrons un graphique représentant le résultat sur Streamlit.

- Quel est le top 10 de job titles les plus postés ?

The screenshot shows the Snowflake interface with the database set to TP_BI and the worksheet titled CITIBIKE_ONE. The code editor contains a query to select job titles and their counts from the JOBS_POSTING table, ordered by count in descending order and limited to 10 rows. The results table shows the top 10 job titles and their respective counts.

TITLE	JOB_COUNT
Sales Director [Owner/Operator]	120
Sales Manager	99
Project Manager	97
Customer Service Representative	85
Senior Accountant	83
Retail Sales Associate	82
Sales Associate	81
Staff Accountant	78
Administrative Assistant	77
Executive Assistant	72

- Quel est le job titles les mieux rémunéré (tenir compte de la devise) ?

Nous avons regardé les différents types de devises indiquées dans la view salaries et il s'avère qu'il y a qu'une seule devise : « USD »

The screenshot shows the Snowflake interface with the database set to TP_BI and the worksheet titled CITIBIKE_ONE. The code editor contains a query to select the job title and maximum salary from the JOBS_POSTING and SALARIES_VIEW tables, joining them on job_id, filtering for US dollars, and ordering by salary in descending order with a limit of 1 row. The results table shows the single record for the Human Resources Director with a salary of 70,000,090,000.

TITLE	MAX_SALARY
Human Resources Director	7000090000

- Quelle est la répartition des offres d'emploi par taille d'entreprise ?

The screenshot shows a Snowflake interface with the following details:

- Database:** TP_BI (selected)
- Worksheet:** CITIBIKE_ONE
- Code Editor:** A query is being run against the LINKEDIN.PUBLIC schema. The code is as follows:

```

377 SELECT title, MAX(max_salary) AS max_salary
378 FROM jobs_posting
379 GROUP BY title
380 ORDER BY max_salary DESC
381 LIMIT 1;
382
383 SELECT c.company_size, COUNT(*) AS job_count
384 FROM jobs_posting jp
385 JOIN companies c ON jp.company_id = c.company_id
386 GROUP BY c.company_size
387 ORDER BY job_count DESC;
388
389 SELECT ci.industry, COUNT(*) AS job_count
390 FROM ...
    
```

- Results:** The results show the count of jobs by company size.

COMPANY_SIZE	JOB_COUNT
1	9658
2	6003
3	3862
4	3308
5	3021
6	2749
7	2495
8	1449
null	

- Quelle est la répartition des offres d'emploi par type d'industrie ?

The screenshot shows a Snowflake interface with the following details:

- Database:** TP_BI (selected)
- Worksheet:** CITIBIKE_ONE
- Code Editor:** A query is being run against the LINKEDIN.PUBLIC schema. The code is as follows:

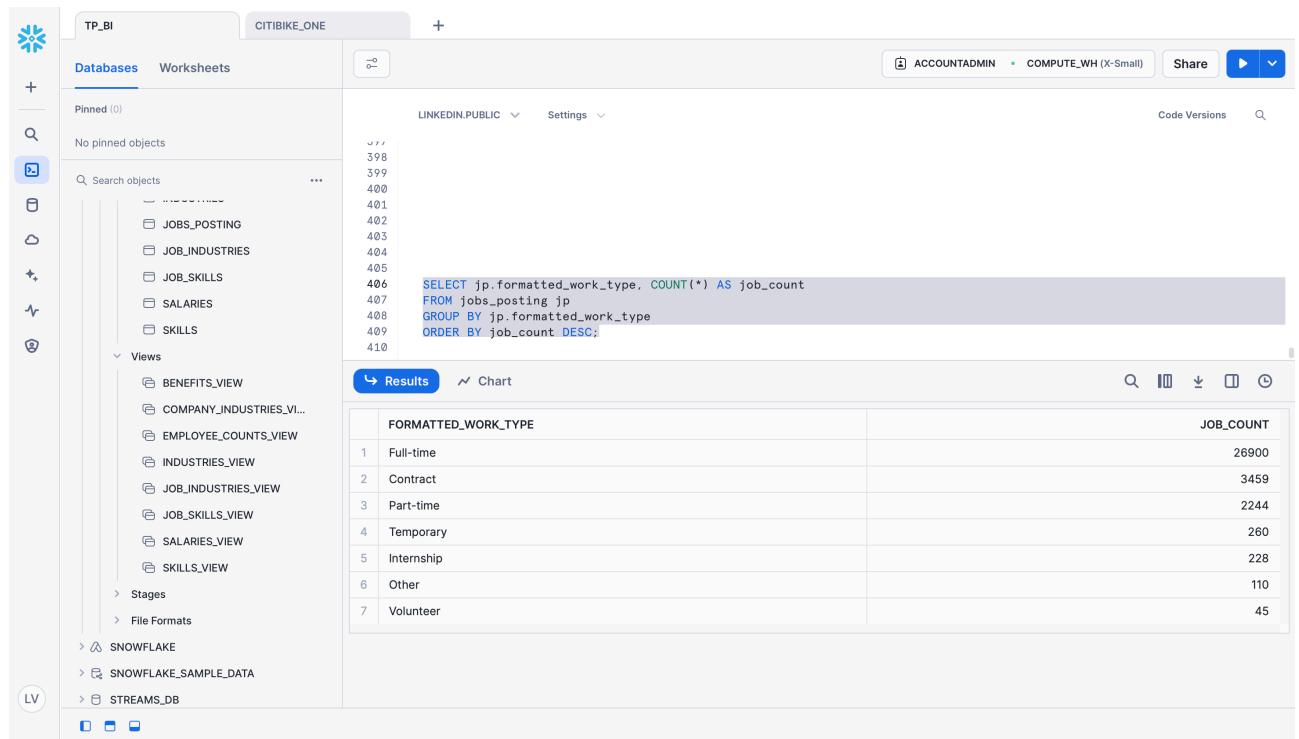
```

385 FROM jobs_posting jp
386 JOIN companies c ON jp.company_id = c.company_id
387 GROUP BY c.company_size
388 ORDER BY job_count DESC;
389
390 SELECT i.industry_name, COUNT(*) AS job_count
391 FROM jobs_posting jp
392 JOIN job_industries_view ji ON jp.job_id = ji.job_id
393 JOIN industries_view i ON ji.industry_id = i.industry_id
394 GROUP BY i.industry_name
395 ORDER BY job_count DESC;
396
397
398 select * from industries ;
399
400
    
```

- Results:** The results show the count of jobs by industry name.

INDUSTRY_NAME	JOB_COUNT
IT Services and IT Consulting	3675
Hospitals and Health Care	3526
Staffing and Recruiting	2730
Retail	1971
Financial Services	1891
Software Development	1599
Construction	1088
Technology, Information and Internet	912

- Quelle est la répartition des offres d'emploi par type d'emploi (full-time, intership, part-time) ?



The screenshot shows the Snowflake UI interface. On the left, there's a sidebar with various icons and a search bar. The main area has tabs for 'TP_BI' and 'CITIBIKE_ONE'. The right side shows a code editor with a query and its results.

```

406   SELECT jp.formatted_work_type, COUNT(*) AS job_count
407   FROM jobs_posting jp
408   GROUP BY jp.formatted_work_type
409   ORDER BY job_count DESC;
410
  
```

Results:

FORMATTED_WORK_TYPE	JOB_COUNT
1 Full-time	26900
2 Contract	3459
3 Part-time	2244
4 Temporary	260
5 Internship	228
6 Other	110
7 Volunteer	45

Vous trouverez ci-dessous la visualisation des précédentes requêtes grâce à Streamlit :

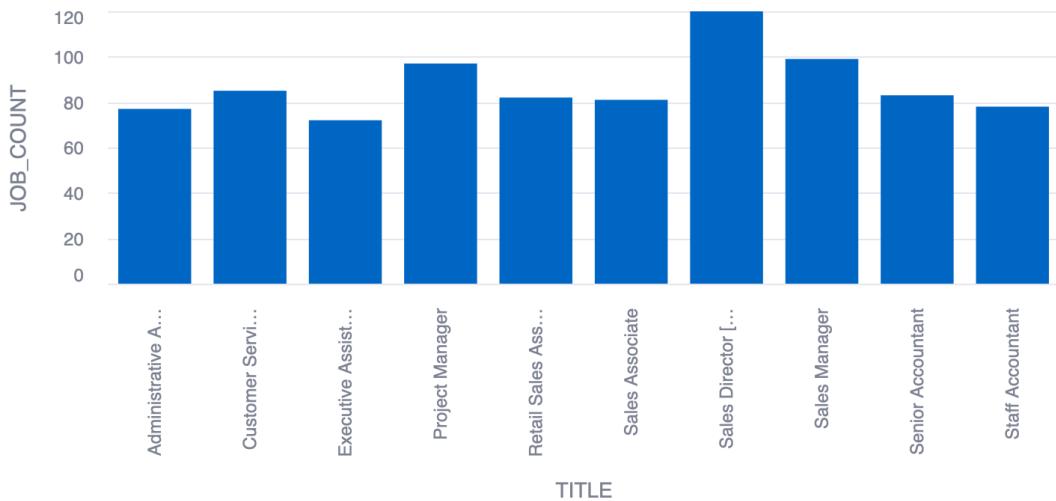
Projet module architecture BI

Laura VERCHERE & Liam HUSSOUD

Classe : 2D

TOP 10 DES JOBS

Quel est le top 10 de job titles les plus postés ?



	TITLE	JOB_COUNT
0	Sales Director [Owner/Operator]	120
1	Sales Manager	99
2	Project Manager	97
3	Customer Service Representative	85
4	Senior Accountant	83
5	Retail Sales Associate	82
6	Sales Associate	81
7	Staff Accountant	78
8	Administrative Assistant	77
9	Executive Assistant	72

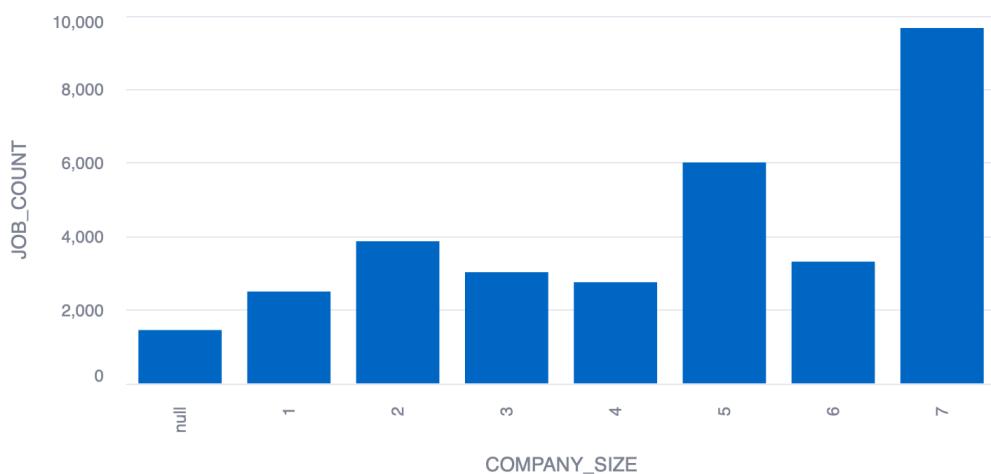
JOB LE MIEUX RÉMUNÉRÉ

Quel est le job le mieux rémunéré (tenir compte de la devise) ? Ici, la devise est en USD.

	TITLE	MAX_SALARY
0	Human Resources Director	7000090000

OFFRE D'EMPLOI PAR TAILLE D'ENTREPRISE

Quelle est la répartition des offres d'emploi par taille d'entreprise ?

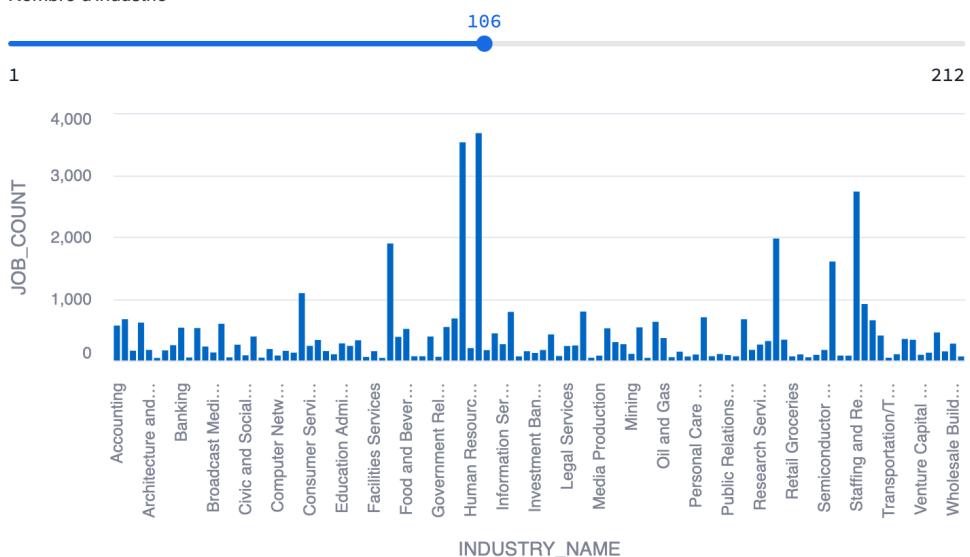


	COMPANY_SIZE	JOB_COUNT
0	7.0000	9658
1	5.0000	6003
2	2.0000	3862
3	6.0000	3308
4	3.0000	3021
5	4.0000	2749
6	1.0000	2127

OFFRE D'EMPLOI PAR TYPE D'INDUSTRIE

Quelle est la répartition des offres d'emploi par type d'industrie ?

Nombre d'industrie



OFFRE D'EMPLOI PAR TYPE D'EMPLOI

Quelle est la répartition des offres d'emploi par type d'emploi (full-time, intership, part-time) ?

