

A lire avant de lire !

Ces exercices sont extraits de [notre programme Data Analyse](#).

Comme vous êtes sur la version gratuite de nos cours, vous n'avez pas accès aux données de Shapr et ne pourrez donc pas les réaliser. Mais ça vous donne un petit aperçu. 😊

Si vous voulez pratiquer avec nous et vraiment apprendre à maîtriser SQL ou en savoir plus sur le programme, vous pouvez prendre RDV avec nous en suivant ce lien : <https://calendly.com/emil-school/rdv-emil>

Exercices introduction au SQL

Vous allez avoir aujourd'hui accès à quasiment toute la totalité de la base de données de la startup Shapr. L'objectif est de prendre en main l'outil Bigquery et d'explorer ensemble le *Data warehouse*.

1) Pourquoi ces requêtes provoquent-elles des erreurs ?

a)

```
SELECT  
FROM shapr.node_human
```

b)

```
SELECT node_idFROM shapr.node_human
```

c)

```
SELECT node_id device  
FROM shapr.node_human
```

d)

```
SELECT *  
WHERE segment > 3
```

```
FROM shapr.node_human
```

e)

```
SELECT node_id  
FROM shapr.node_human  
WHERE device == "ios"
```

f)

```
SELECT node_id  
FROM shapr.node_human  
ORDER 1 DESC
```

g)

```
SELECT node_id  
FROM shapr.node_human  
ORDER 1, 2
```

h)

```
SELECT node_id  
FROM shapr.node_human  
ORDER BY 1, 2
```

i)

```
SELECT node_id  
FROM shapr.node_human  
WHERE email LIKE g%
```

2) Prendre en main BigQuery

Dans Bigquery, notre base Shapr correspond au dossier du même nom qu'on appelle un *dataset*. Au sein de ce *dataset*, vous trouverez toutes les tables. Pour y accéder, cliquez sur ce [lien](#).

- a) Identifiez le projet data-analytics-essentials-sc dans lequel vous êtes rattaché.

- b) Identifiez le nombre de datasets disponibles (un dataset = une base de données)
- c) Observez le nombre de tables disponibles dans le dataset Shapr
- d) Cliquez sur une table et observez les informations qui vous sont affichées sous l'encadré de création de requêtes
- e) Pouvez-vous observer les types de chaque colonne ?
- f) Regardez un aperçu des données de la table sélectionnée
- g) Pouvez-vous voir le nombre de lignes et de colonnes par table ?

3) Qui sont nos utilisateurs V1 ?

Chaque utilisateur est identifié en base par son id dans l'application qui correspond à la colonne **node_id**. Nous allons utiliser la table **node_human** pour les questions qui vont suivre. Écrivez les requêtes correspondantes:

- a) Affichez toutes les colonnes de la table.
- b) Affichez toutes les colonnes de la table et seulement 50 lignes.
- c) Affichez seulement les node_id utilisateurs de ces 50 lignes.
- d) Affichez les node_id des utilisateurs et leurs email qu'ils utilisent.
- e) Etudiez la clause **DISTINCT**. Combien de mails différents sont-ils utilisés par les utilisateurs ? (le nombre de lignes est indiqué au niveau des résultats)
- f) Affichez les node_id des utilisateurs de gmail uniquement
- g) Affichez les node_id et l'industry des utilisateurs ayant instagram, une photo de profil et basés en france
- h) Affichez les informations de 50 utilisateurs de Montréal au Québec

city : "Montréal area"

- i) Retournez les 10 dernières personnes à s'être inscrites en Belgique. La colonne **member_at** donne la date d'inscription de l'utilisateur.

country: "Belgium"

- j) Pour un besoin d'analyse, affichez tous les node_id utilisateurs qui se sont inscrits en janvier 2020 et qui ont un segment utilisateur 8. Pour faciliter la compréhension, on souhaite que la colonne en sortie soit renommée en "id" et non pas "node_id". Avez-vous des résultats ?

Pour créer une condition sur une date, on peut utiliser les mêmes opérateurs que vus en cours (>, <, >=, <=) avec des dates écrites sous ce format "2019-02-28". SQL va interpréter "2019-02-28" comme "2019-02-28 00:00:00", c'est-à-dire à minuit pile.

- k) En cours, nous n'avons pas vu une valeur fondamentale en SQL qu'il est important de connaître. Que fait cette requête selon vous ?

```
SELECT DISTINCT node_id
FROM shapr.node_human
WHERE industry IS NULL
```

A quoi fait référence le NULL ici ?

Un petit tour [ici](#) pour avoir plus d'explications.

- l) Ecrivez une requête pour trouver toutes les villes uniques qui commencent par "New". Écrivez-en une seconde mais en utilisant la nouvelle clause [SUBSTR](#) en regardant sa documentation.

Cette clause pourra être utilisée dans la clause WHERE

4) Qui sont nos utilisateurs V2 ?

- a) On s'intéresse aux utilisateurs Américains qui n'ont pas un profil professionnel rempli. L'équipe Marketing a défini pas "rempli" par :
- soit ils n'ont pas de photo
 - soit ils n'ont **pas** de site perso et **pas** de linkedin.

Aidez l'équipe Marketing à identifier ces utilisateurs !

b) Trouvez toutes les villes uniques des utilisateurs américains de l'état d'Ohio

c) On vient vous voir et on vous dit:

"Est-ce que tu pourrais me retrouver l'id de cette personne stp ?

*J'ai les caractéristiques de son profil mais plus son id. Il est **segment 6**, il m'a écrit en utilisant une **adresse gmail**, quand j'ai regardé son compte, son téléphone utilisait en **langue française** et il est sur **iphone**. Il vient de **Paris**. Il s'est inscrit **en octobre 2017** et a '**Computer Software**' comme industrie."*

Pouvez l'identifier de façon certaine ?

d) Comptez le nombre d'utilisateurs n'ayant pas de photo, pas de site perso et aucune présence sur les réseaux sociaux

Dans BigQuery, vous pouvez voir le nombre de résultats de votre requête en base de page.

e) Quelle est la proportion de ces personnes en France ?

On fera deux requêtes et on notera le nombre de personnes à chaque fois pour faire le rapport ensuite à la main

f) La proportion est-elle plus importante aux Etats-unis ?

g) On va s'intéresser maintenant à la proportion des utilisateurs dans les grandes villes par pays. Y-a-t'il des gens qui sont au United Kingdom mais qui n'ont pas de ville en base ? Combien de personnes sont dans ce cas ?

h) Quelle est la proportion d'utilisateurs au United Kingdom ayant une ville enregistrée en base et qui **ne sont pas dans les areas suivantes:**

- London area
- Manchester area
- Birmingham area

i) En est-il de même pour Paris versus le reste de la France ?

5) [Bonus] Création de bases de données et de tables

Si vous vous demandez comment sont fabriquées les bases de données et les tables, lisez l'article [suivant](#) pour le découvrir jusqu'au chapitre ALTER TABLE !
On utilise en effet d'autres requêtes SQL pour le faire !

6) Discussion de groupe: Tables, Vues, Vues matérialisées

Lisez le début de l'article [suivant](#) sur ce qu'est une view (jusqu'à complex view, le reste est hors programme pour le moment). Nous discuterons tous ensemble de l'intérêt d'utiliser des vues à la fois pour l'accès aux données mais aussi pour les besoins de reporting.

7) Exercices HackerRank

[HackerRank](#) est une plateforme de pratique du code extrêmement populaire. Certains recruteurs font même passer des coding challenges dessus.

Créez vous donc un compte et pratiquez les exercices suivants (NB : N'oubliez pas de mettre l'éditeur de code MySQL avant d'écrire vos clauses) :

- a) [Weather Observation Station 3](#)
- b) [Weather Observation Station 4](#) (pour cette clause regardez la documentation pour [COUNT\(\)](#))
- c) [Weather Observation Station 6](#)
- d) [Weather Observation Station 8](#)
- e) [Weather Observation Station 11](#)

Si vous avez tout fini, bravo ! Vous pouvez continuer à faire des challenges sur HackerRank (attention certains exercices demandent des notions que nous n'avons pas encore vu).