

Introduction: Bases de Données Master Digital Marketing

Pratique avec [BigQuery](#)

Présenté par
Kevin Rosamont

Qu'est-ce que le SQL?

SQL

SQL veut dire Structured Query Language

Language permettant de manipuler les données au sein des bases de données

Nous allons utiliser le SQL dans ce cours uniquement pour lire les données

**Pourquoi vous apprendre
à utiliser SQL**

Pourquoi apprendre à coder?

Les entreprises dans le digital ont souvent des bases de données

Interroger les données pour une meilleure connaissance clients

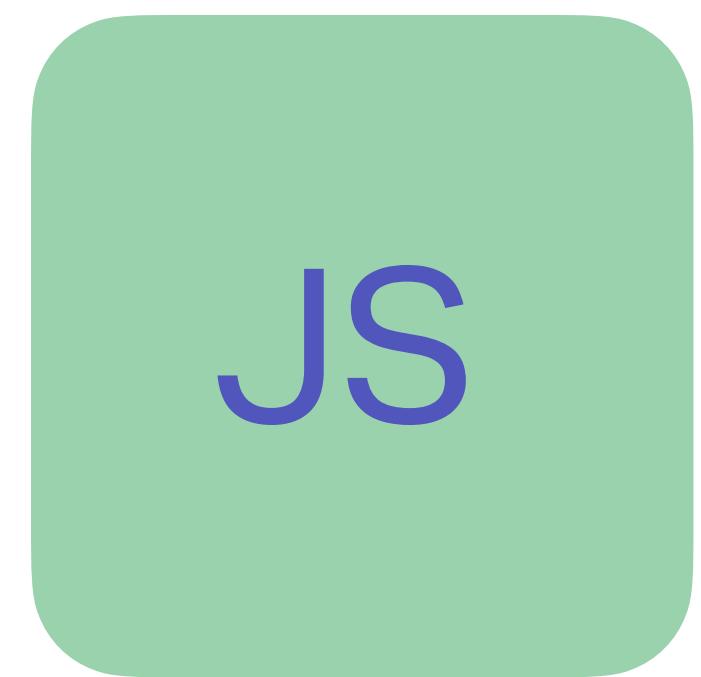
Meilleur suivi de l'évolution de votre activité
(en utilisant SQL et data studio)

Avoir un avantage compétitif!

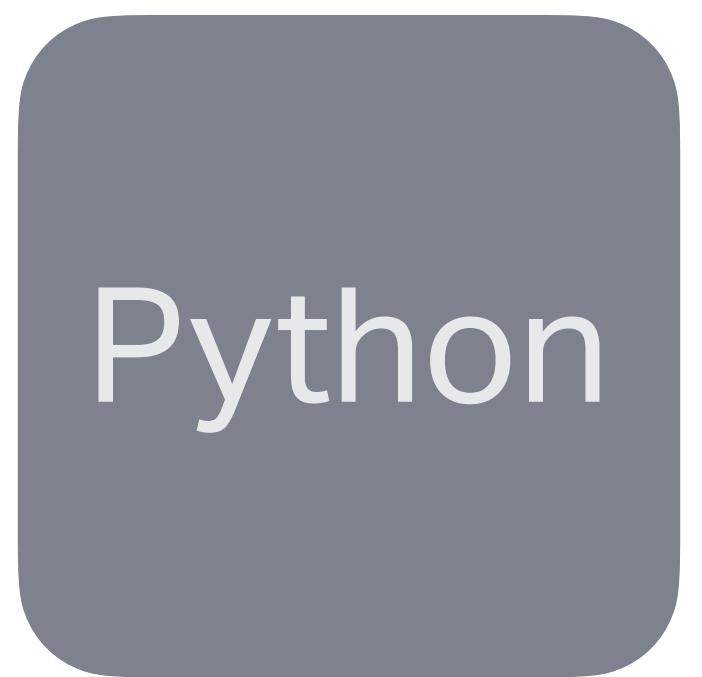
Les langages populaires



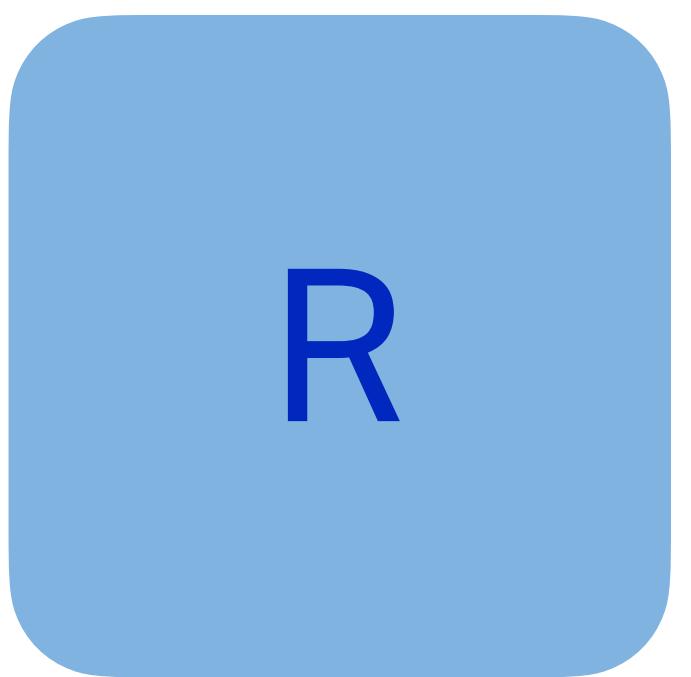
Page web
Contenu
Style



Animation web
WebApp
MobileApp



Automatisation
Deep Learning
Test Sécurité



Data Analysis
Visualisation
Automatisation



Base de données
Interroger, créer
Modifier

Notre but durant ce module

Ajouter un esprit data driven à vos compétences!

Connaitre vos clients

Savoir ce qui fonctionne dans vos campagnes marketing

Proposer de nouvelles offres

Tout cela grâce à la data

Ce que nous allons voir

Learning by doing!

Introduction aux bases de données relationnelles

Introduction au langage SQL

Mise en pratique sur des données E-commerces

Introduction aux bases de données relationnelles

Qu'est-ce que les données?

Donnée :

Représentation d'une information sous un format digital

Texte

Valeur numériques

Image

Sons

(des formats possibles)

Qu'est-ce que les données?

Donnée :

Nous allons principalement travailler avec les formats suivants

INTEGER, DECIMAL, NUMERIC : chiffres avec ou sans décimal

STRING : format textes

BOOLEAN : représentées par les mots clés **TRUE** et **FALSE** (non sensibles à la casse)

TIMESTAMP, DATE, TIME : format date et heure

GEOGRAPHY : lieu géographique

Plusieurs type de données

Structurées

Les données généralement facile à lire pour les humains, tels que Google Sheets ou Excel

Semi-Structurées

Forme de données structurées qui n'obéit pas à la structure formelle des modèles de tableaux de données mais qui comportent des informations associées, des métadonnées : Email, XML, NoSQL, JSON

Non-Structurées

Représentent la plupart des données, c'est des données qui ne peuvent pas être classées : Video, Sons, Image ou Text

Qui crée la donnée?

Les humains

Quand un Email est envoyé ou un tableau Excel est créé, de la donnée est générée.

Les machines

Quand une page internet est visité, Quand la météo est enregistrée, de la donnée est créée par des machines.

Qu'est-ce qu'une base de données?

Une base de données est un ensemble de données structuré et organisé permettant le stockage d'informations

Elle évite au maximum les réplications de données (perte de place, source d'erreurs) ou redondance contrôlée

Les données doivent être interrogables selon leur contenu

SGBDR: Système de Gestion de Base de Données Relationnelle

Logiciels permettant l'utilisation des Bases de Données,
Ils ont pour objectifs:

Manipulation des données

Cohérence des données

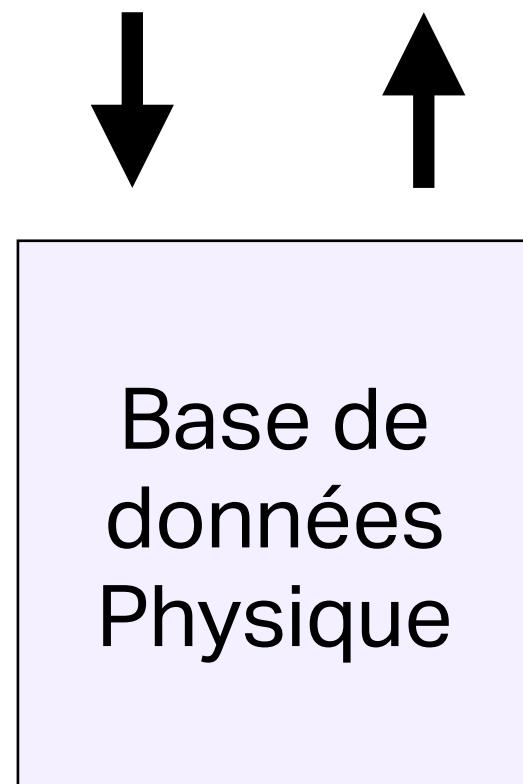
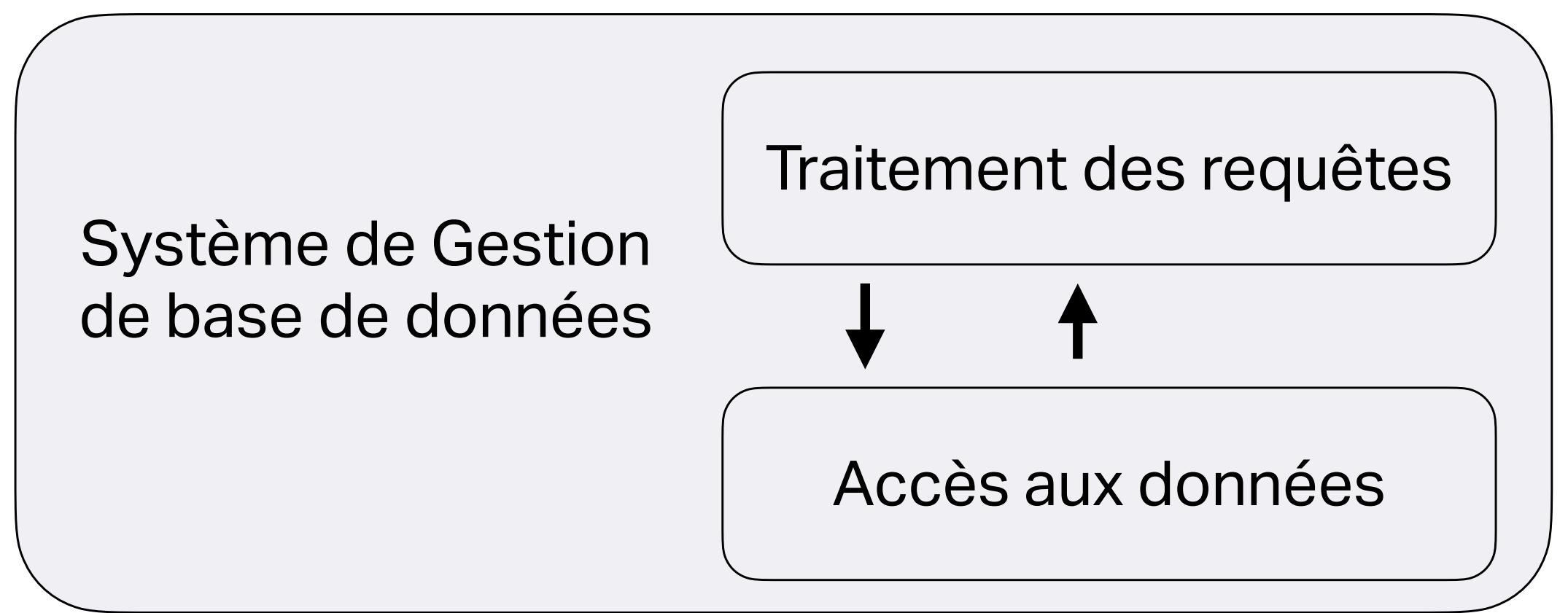
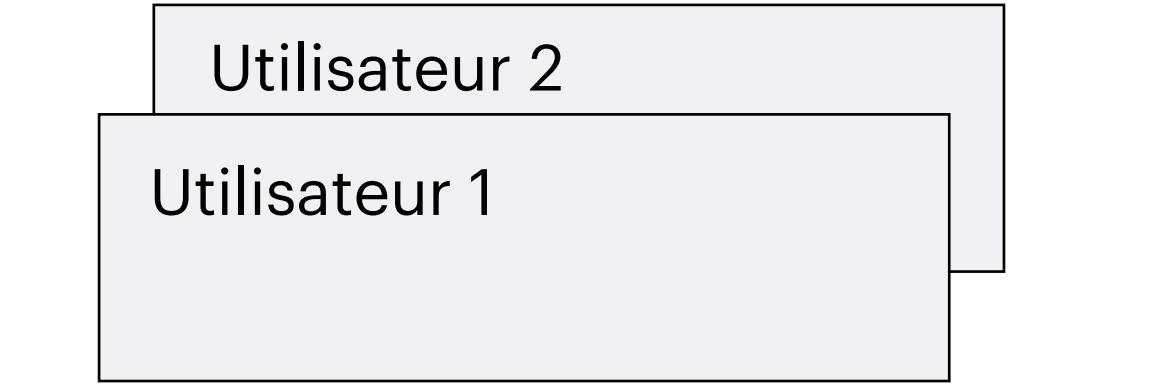
Fournir un accès efficace aux données

Partage des données

Contrôler la redondance des données

Sécurité des données

Fonctionnement



Niveau Conceptuel

Exemple de table

Les colonnes sont appelées les **attributs**

Les lignes sont appelées les **tuples**

station_id	name	status	address	council_district
3464	Pease Park	closed	1155 Kingsbury St	9
2500	Republic Square	closed	425 W 4th Street	9
2536	Waller & 6th St.	closed	602 Waller St.	3
2538	Bullock Museum @ Congress & MLK	closed	1881 Congress Ave.	1
2541	State Capitol @ 14th & Colorado	closed	206 W. 14th St.	1
2545	ACC - Rio Grande & 12th	closed	700 W. 12th St.	9
2546	ACC - West & 12th Street	closed	1231 West Ave.	9
2550	Republic Square @ Guadalupe & 4th St.	closed	Presented by Austin Ventures	9

Clef primaire

Chaque table doit avoir une **clef primaire** constituée par un ensemble minimum d'attributs permettant de distinguer les lignes

Chaque ligne est **unique**

Il y a des cas où plusieurs clefs primaires sont possibles, on parle de **clefs candidates**, il faut choisir sa **clef primaire**

station_id	name	status	address	council_district
3464	Pease Park	closed	1155 Kingsbury St	9
2500	Republic Square	closed	425 W 4th Street	9
2536	Waller & 6th St.	closed	602 Waller St.	3
2538	Bullock Museum @ Congress & MLK	closed	1881 Congress Ave.	1
2541	State Capitol @ 14th & Colorado	closed	206 W. 14th St.	1
2545	ACC - Rio Grande & 12th	closed	700 W. 12th St.	9
2546	ACC - West & 12th Street	closed	1231 West Ave.	9
2550	Republic Square @ Guadalupe & 4th St.	closed	Presented by Austin Ventures	9

Clef étrangère

Le lien entre 2 tables se fait grâce à une **clef étrangère**

Une clef étrangère est une clef primaire d'une autre table qui n'est plus forcément unique.

station_id	name	status	address	council_district
3464	Pease Park	closed	1155 Kingsbury St	9
2500	Republic Square	closed	425 W 4th Street	9
2536	Waller & 6th St.	closed	602 Waller St.	3
2538	Bullock Museum @ Congress & MLK	closed	1881 Congress Ave.	1
2541	State Capitol @ 14th & Colorado	closed	206 W. 14th St.	1
2545	ACC - Rio Grande & 12th	closed	700 W. 12th St.	9
2546	ACC - West & 12th Street	closed	1231 West Ave.	9
2550	Republic Square @ Guadalupe & 4th St.	closed	Presented by Austin Ventures	9

Exemple: Achat de produit sur Amazon



Exemple: Facturation de l'achat de produits

Clients
- email - nom - prenom - date_inscription - anniversaire - adresse

Produits
- id_produit - nom - categorie - sous-categorie - couleur - dimension - poids - prix

clef primaire

clef étrangère

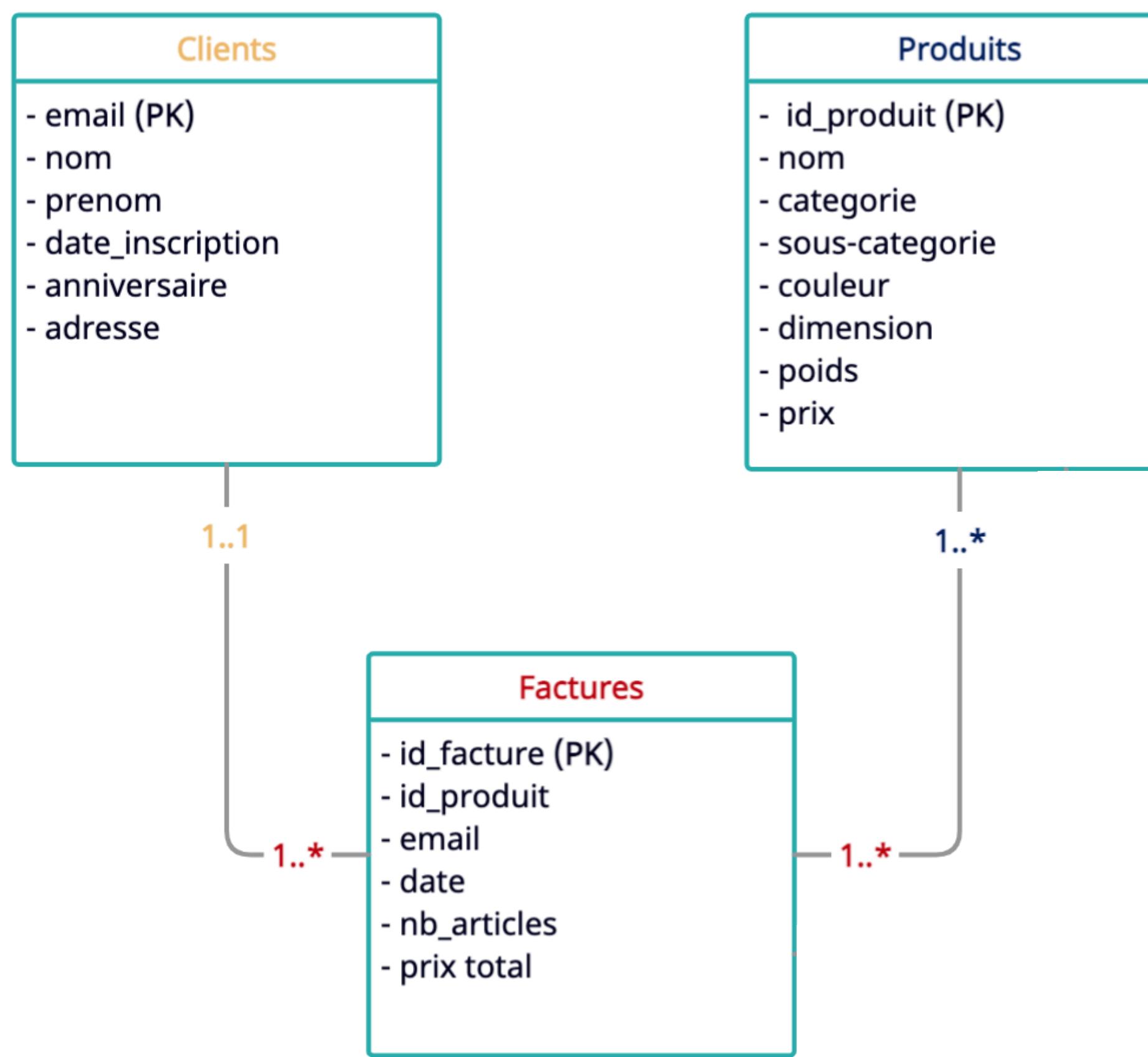
Exemple: Facturation de l'achat de produits

3 grands types de liens :

1..1 à 1..1 (un à un)

1..1 à 1..* (un à plusieurs)

1..* à 1..* (plusieurs à plusieurs)



Se lit :

Un **client** peut avoir **1..*** **facture(s)**

Une **facture** peut avoir **1..*** **produit(s)**

Exemple: Facturation de l'achat de produits

Vérifions que notre schéma soit bien construit

Création d'observation dans Excel

Produits
- id_produit (PK)
- nom
- categorie
- sous-categorie
- couleur
- dimension
- poids
- prix

Factures
- id_facture (PK)
- id_produit
- email
- date
- nb_articles
- prix total

A retenir de l'exemple

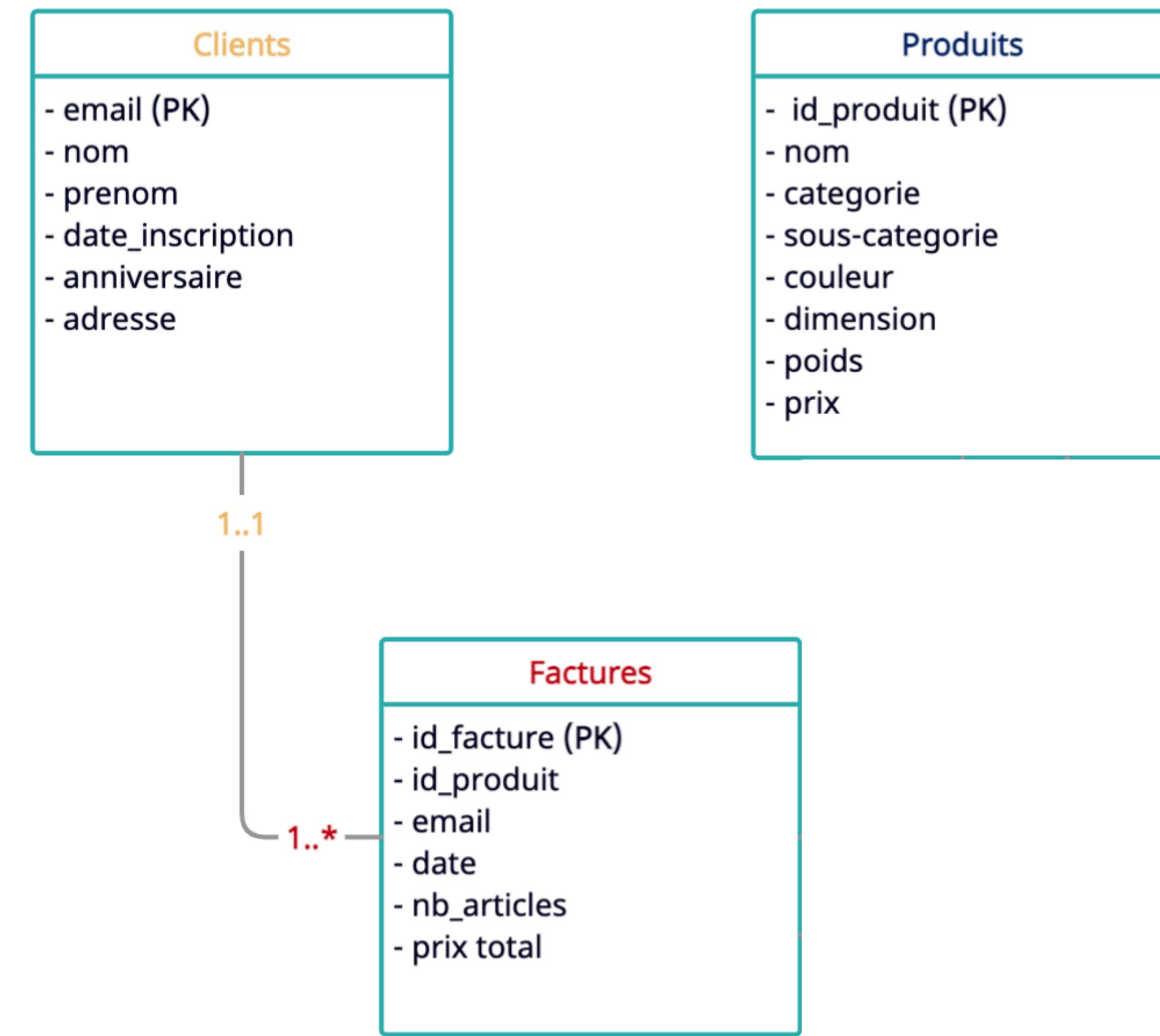
A chaque fois que vous avez
une relation 1..* à 1..*
il faut créer une d'association

Passer à 2 relations 1..1 à 1..*

Se lit :

Un client peut avoir 1..* facture(s)

Une facture peut être dans 1..*
association factureproduits

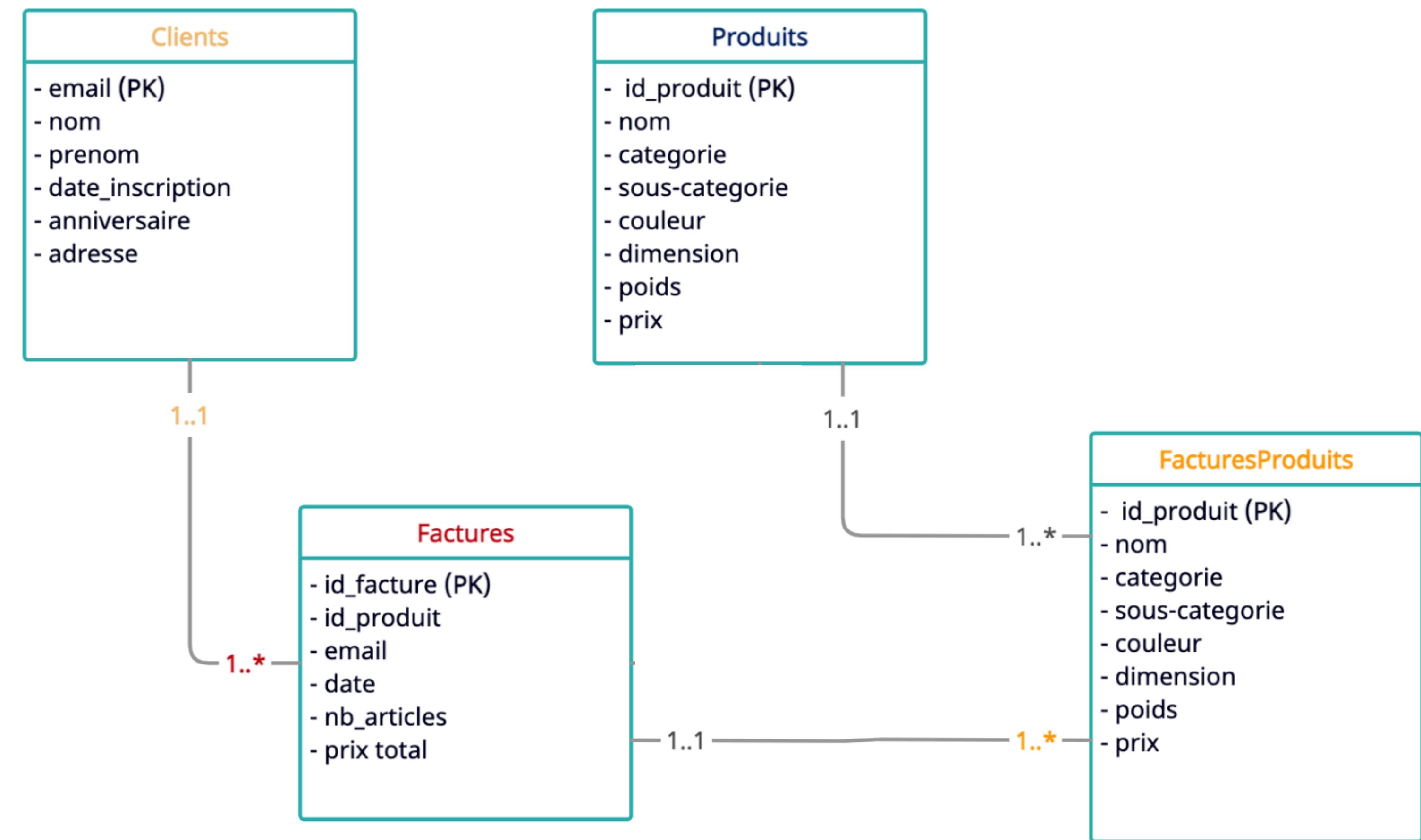


A retenir de l'exemple

A chaque fois que vous avez
une relation 1..* à 1..*
il faut créer une d'association

Passer à 2 relations 1..1 à 1..*

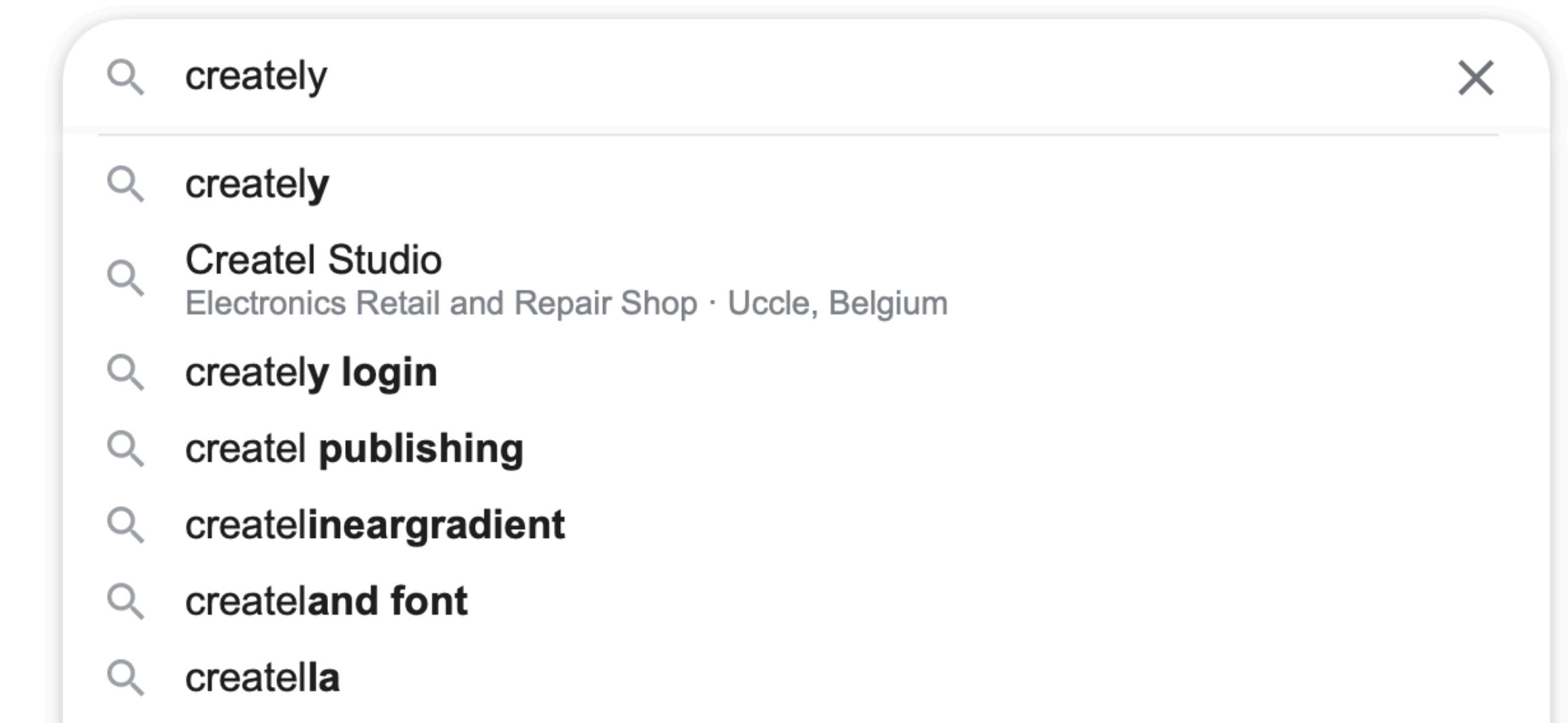
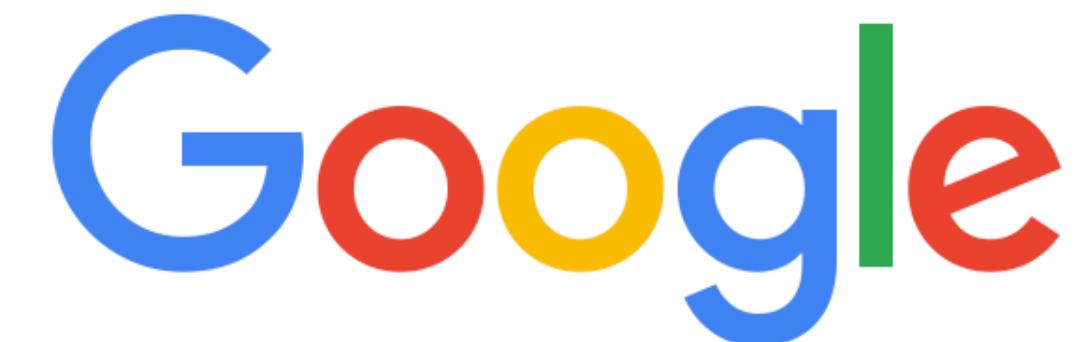
Se lit :
Une facture peut être sur 1..*
association factureproduits



**Utilisation de la webApp
creately (gratuit) pour
faire les schémas**

Faisons un exemple ensemble!

Site pour faire des schémas relationnels
<https://app.creately.com/>



A votre tour!

5 groupes, 3 exemples de schémas relationnels
correspondants au fonctionnement de votre entreprise

Création des 5 rooms!

Le SQL pour les digital Marketers

Qu'est-ce que le SQL?

Structured Query Language

Langage de Programmation crée chez IBM
au début des années 1970 (E. Codd)

Gérer un système de gestion de base de données
relationnelles

3 sous-langages SQL

Data Definition Language (DDL):

Manipulation des structures des données

Data Manipulation Language (DML):

Créer, Lire, Mettre à jour, Supprimer (CRUD)

Data Control Language (DCL):

Contrôler l'accès aux données dans la database

En tant que Digital Marketer, Qu'avez-vous besoin de connaître?

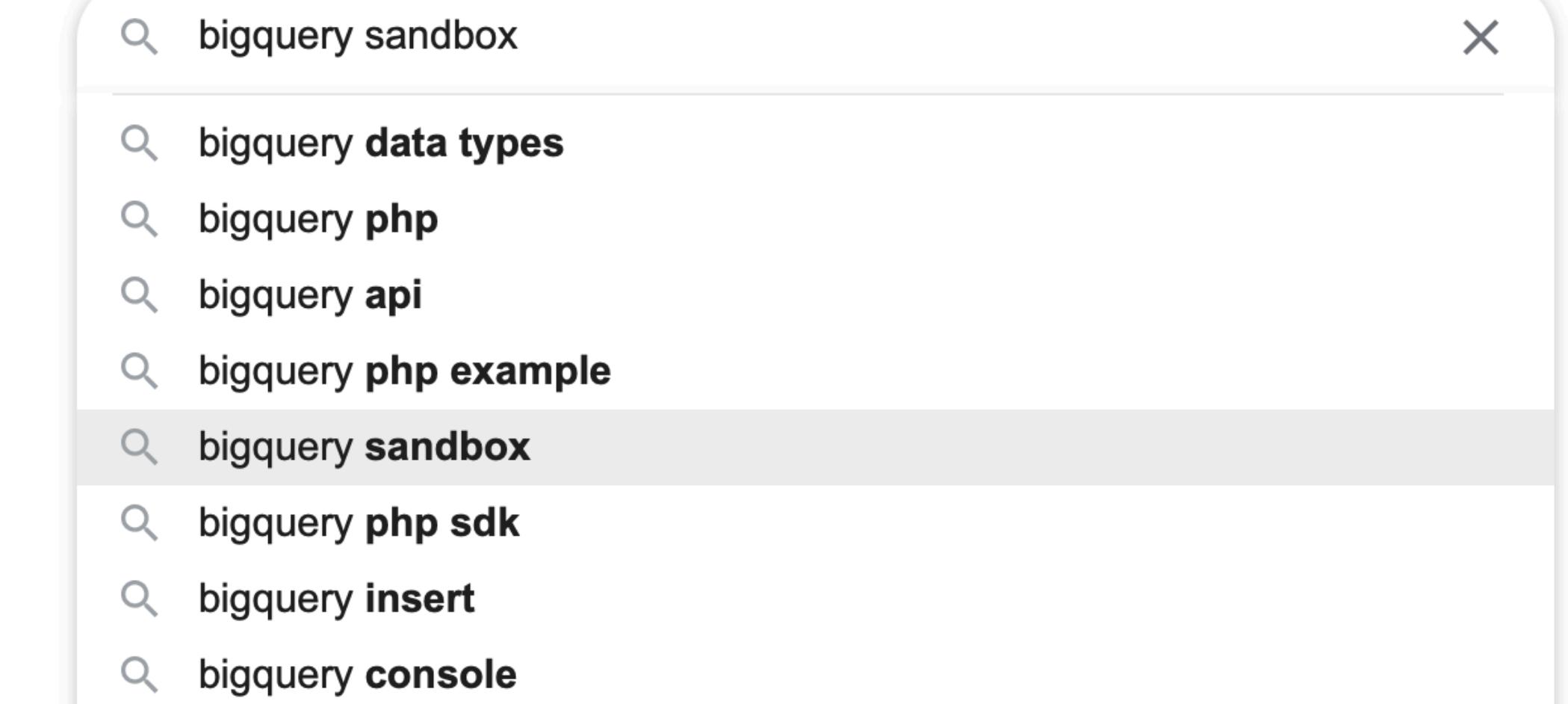
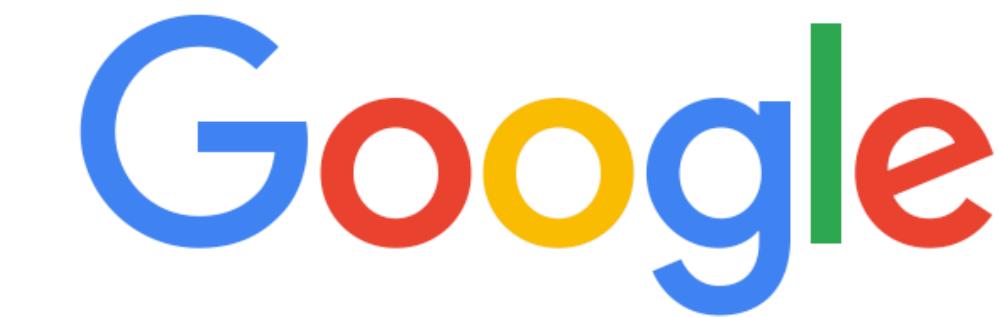
CREATE
READ
UPDATE
DELETE

READ
va permettre d'interroger les données
pour calculer vos KPI,
mettre en valeurs certaines informations.

Ouverture de compte BigQuery (gratuit)

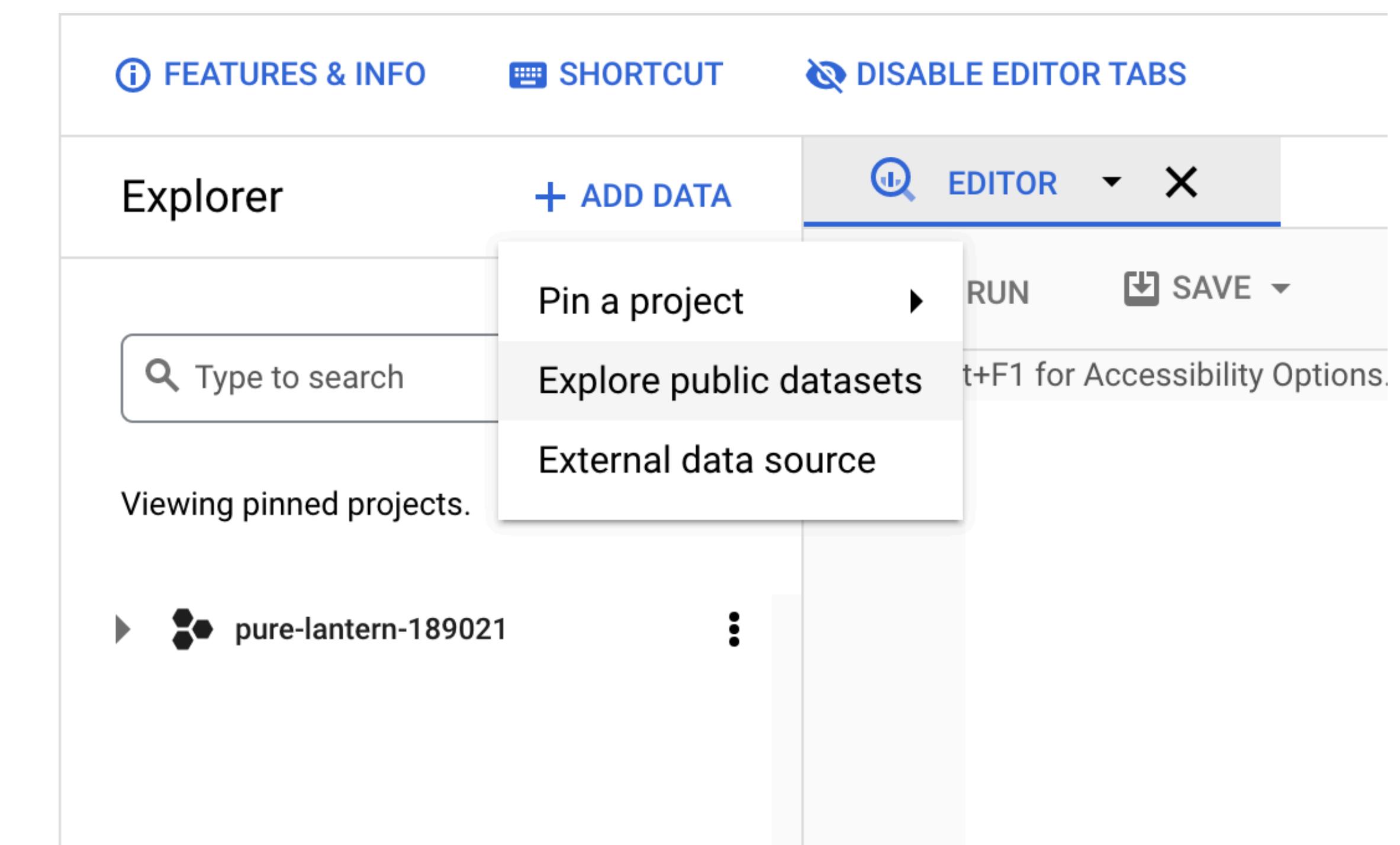
Utilisation de BigQuery

[Allez dans BigQuery Sandbox](#)



Accéder aux Données

1- Allez dans Explore public datasets



Accéder aux Données

2- Taper "e commerce" (barre de recherche)

The screenshot shows a search interface for a marketplace. At the top, there is a header with a shopping cart icon labeled "Marketplace" and a search bar containing the text "e commerce". Below the header, a breadcrumb navigation shows "Marketplace > 'e commerce' > Datasets". A filter bar with a "Filter" button and a placeholder "Type to filter" is visible. The main area is titled "Datasets" and contains a table with two columns: "Category" and "Type". The "Category" column lists "Analytics", "Machine learning", "Healthcare", "Climate", "Datasets for COVID-19 research", and "Financial services", each with a count of "(2)", "(1)", "(1)", "(1)", "(1)", and "(1)" respectively. The "Type" column lists "Datasets" and "X". Two dataset entries are fully visible: "Iowa Liquor Retail Sales" from the Iowa Department of Commerce, which describes wholesale liquor purchases in Iowa, and "Severe Storm Event Details" from NOAA, which describes severe weather events across the US. Both entries include small circular logos.

Category	Type
Analytics	Datasets
Machine learning	X
Healthcare	
Climate	
Datasets for COVID-19 research	
Financial services	

Iowa Liquor Retail Sales
Iowa Department of Commerce

This dataset contains every wholesale purchase of liquor in the State of Iowa by retailers for sale to consumers. The Iowa Alcoholic Beverage Control Board controls the wholesale distribution of liquor intended for retail sale, which means this dataset covers all sales of liquor in the entire state. The dataset contains every wholesale order of liquor by all grocery stores, liquor stores, restaurants, bars, and other retail outlets in the state.

Severe Storm Event Details
NOAA

The Storm Events Database is an integrated database of severe weather events across the United States. It provides detailed information about a storm event's location, azimuth, distance, impact, and severity, including the cost of damage and fatalities. The database is used for documenting the occurrence of storms and other significant weather phenomena having substantial impact on the nation.

Accéder aux Données

3- Sélectionner Iowa Liquor Retail Sales

Marketplace X

Marketplace > "e commerce" > Datasets

Filter Type to filter

Datasets

Category	^	3 results
Analytics	(2)	
Machine learning	(1)	
Healthcare	(1)	
Climate	(1)	
Datasets for COVID-19 research	(1)	
Financial services	(1)	

Iowa Liquor Retail Sales
Iowa Department of Commerce

This dataset contains every wholesale purchase of liquor in the State of Iowa by retailers for sale. The Iowa Department of Revenue controls the wholesale distribution of liquor intended for retail sale, which means this dataset covers all sales of liquor in the entire state. The dataset contains every wholesale order of liquor by all grocery stores, liquor stores, and restaurants in the state.

Severe Storm Event Details
NOAA

The Storm Events Database is an integrated database of severe weather events across the United States. It provides detailed information about a storm event's location, azimuth, distance, impact, and severity, including the cost of damage and fatalities. The database is used for documenting the occurrence of storms and other significant weather phenomena having substantial impact on the nation.

Accéder aux Données

4- Cliquer sur VIEW DATASET



Iowa Liquor Retail Sales

Iowa Department of Commerce

Liquor sales in Iowa since 2012, by store, by item, and by day

[VIEW DATASET](#)

Click to view dataset

[OVERVIEW](#)

[SAMPLES](#)

Overview

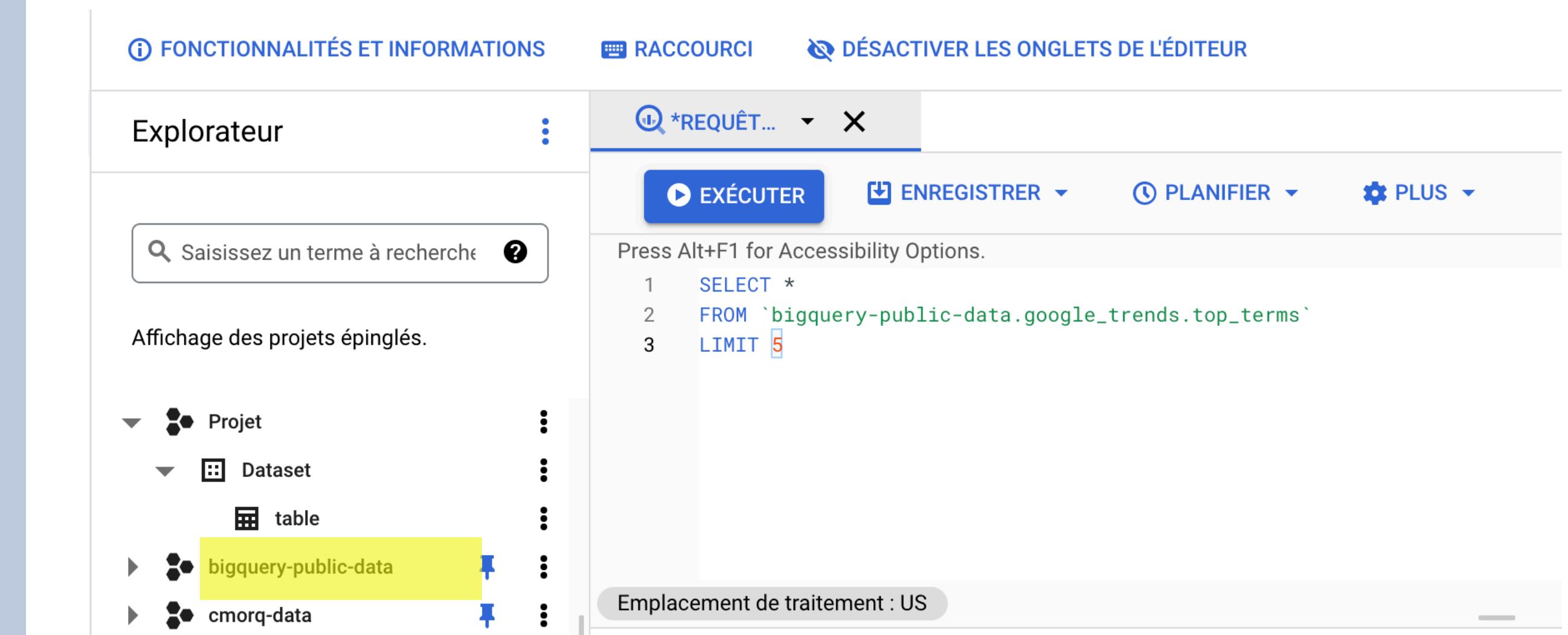
This dataset contains every wholesale purchase of liquor in the State of Iowa by retailers for sale to individuals since January 1, 2012. The State of Iowa controls the wholesale distribution of liquor intended for retail sale, which means this dataset offers a complete view of retail liquor sales in the entire state. The dataset contains every wholesale order of liquor by all grocery stores, liquor stores, convenience stores, etc., with details about the store and location, the exact liquor brand and size, and the number of bottles ordered.

In addition to being an excellent dataset for analyzing liquor sales, this is a large and clean public dataset of retail sales data. It can be used to explore problems like stockout prediction, retail demand forecasting, and other retail supply chain problems.

The [data dictionary](#) is available from the State of Iowa's [Alcoholic Beverages Division](#), within the Iowa Department of Commerce . There are

Accéder aux Données

5- Nouveau projet disponible



The screenshot shows the BigQuery interface. On the left, the 'Explorateur' sidebar lists 'Projet' (Project) and 'Dataset'. Under 'Dataset', there are two entries: 'table' and 'bigquery-public-data'. The 'bigquery-public-data' entry is highlighted with a yellow background. On the right, the main area displays a query editor titled '*REQUÊT...'. The query text is:

```
1 SELECT *  
2 FROM `bigquery-public-data.google_trends.top_terms`  
3 LIMIT 5
```

The status bar at the bottom indicates 'Emplacement de traitement : US' (Treatment location: US).

Accéder aux Données

6- Attributs et format

bikeshare_trips		COPY	SNAPSHOT	DELETE	1
SCHEMA	DETAILS	PREVIEW			
Field name	Type	Mode	Policy Tags	?	Description
trip_id	INTEGER	NULLABLE			Numeric ID of bike trip
subscriber_type	STRING	NULLABLE			Type of the Subscriber
bikeid	STRING	NULLABLE			ID of bike used
start_time	TIMESTAMP	NULLABLE			Start timestamp of trip
start_station_id	INTEGER	NULLABLE			Numeric reference for start station
start_station_name	STRING	NULLABLE			Station name for start station
end_station_id	STRING	NULLABLE			Numeric reference for end station
end_station_name	STRING	NULLABLE			Station name for end

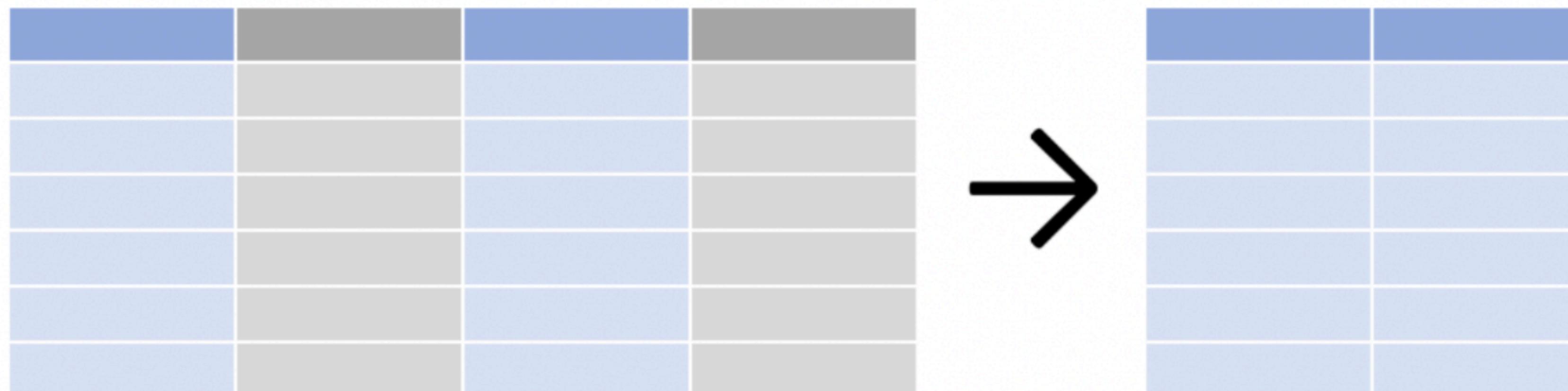
Les requêtes SQL

Structure d'un query SQL

```
SELECT [DISTINCT] column, AGG_FUNC(column_or_expression), ...
FROM mytable
WHERE constraint_expression
GROUP BY column
HAVING constraint_expression
ORDER BY column ASC/DESC
LIMIT count;
```

Manipulation de données via SQL

SELECT (filtre sur les attributs)



Manipulation de données via SQL

4 mots :

SELECT
FROM

Sélectionner les colonnes
Provenance de la colonne

Manipulation de données via SQL

4 mots :

SELECT

Sélectionner les colonnes

FROM

Provenance de la colonne

[ORDER BY]

Ordonner en fonction d'une colonne

[LIMIT]

Nombre de lignes à montrer

Exemple :

```
SELECT station_id as identifiant, name as nom
```

```
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`
```

```
LIMIT 10
```

Utilisation de labels

Avec le mot clef `as`, on peut changer le nom des éléments

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
ORDER BY nom [DESC]  
LIMIT 10
```

Sélectionner l'ensemble des colonnes

Avec SELECT * toutes les colonnes sont sélectionnées

Exemple :

```
SELECT *
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`
LIMIT 10
```

Utilisation de labels

Avec le mot clef `as`, on peut changer le nom des éléments

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
LIMIT 10
```

Afficher des observations distinctes

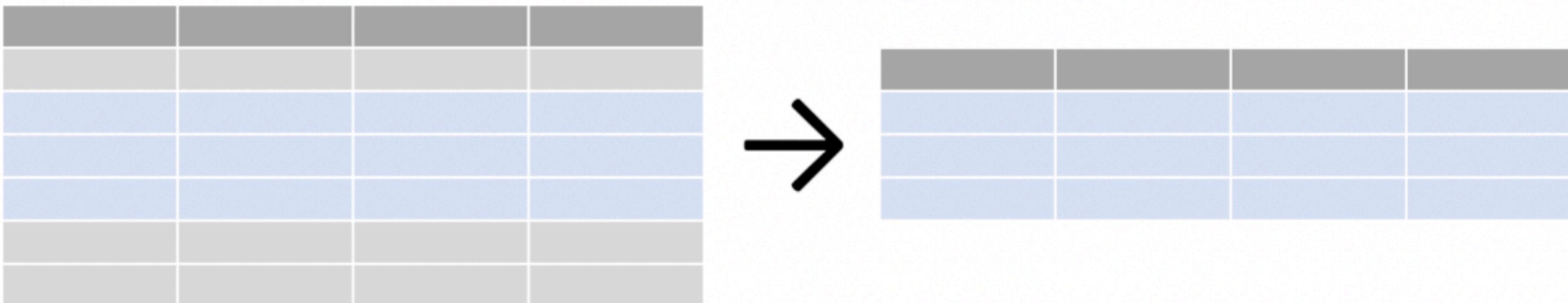
SELECT DISTINCT permet d'afficher **des lignes différentes**

Exemple :

```
SELECT DISTINCT name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
LIMIT 100
```

Manipulation de données via SQL

WHERE (filtre sur les observations)



Utilisation de WHERE

Avec WHERE plusieurs conditions peuvent être réalisées:

=, >, <, >=, <=, <>

LIKE, NOT, IN, BETWEEN AND

OR, AND

IS NULL, IS NOT NULL

Utilisation de WHERE

Avec WHERE plusieurs conditions peuvent être réalisées:

=, >, <, >=, <=, <>

LIKE, NOT, IN, BETWEEN AND

OR, AND

IS NULL, IS NOT NULL

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
WHERE status = "closed"  
LIMIT 100
```

Utilisation de WHERE

Avec WHERE plusieurs conditions peuvent être réalisées:

>, <, >=, <=, <>

LIKE, NOT, IN, BETWEEN AND

OR, AND

IS NULL, IS NOT NULL

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
WHERE status <> "closed" AND council_district = 9  
LIMIT 100
```

Utilisation de WHERE

Avec WHERE plusieurs conditions peuvent être réalisées:

>, <, >=, <=, <>

LIKE, NOT, IN, BETWEEN AND

OR, AND

IS NULL, IS NOT NULL

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
WHERE council_district BETWEEN 2 AND 10  
LIMIT 100
```

Utilisation de WHERE

Avec WHERE plusieurs conditions peuvent être réalisées:

>, <, >=, <=, <>

LIKE, NOT, IN, BETWEEN AND

OR, AND

IS NULL, IS NOT NULL

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
WHERE council_district IN (2,6,7,9)  
LIMIT 100
```

Utilisation de WHERE

Avec WHERE plusieurs conditions peuvent être réalisées:

>, <, >=, <=, <>

LIKE, NOT, IN, BETWEEN AND

OR, AND

IS NULL, IS NOT NULL

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
WHERE notes IS NOT NULL  
LIMIT 100
```

Description

Operateur	Description
+	Effectuer une addition
-	Effectuer une soustraction
*	Effectuer une multiplication
/	Effectuer une division

Description

Operateur	Description
<	Vérifier que la partie à gauche soit plus petite que la partie à droite du signe < Exemple: WHERE price < 20
<=	Vérifier que la partie à gauche soit plus petite ou égale à la partie à droite du signe < Exemple: WHERE price <= 20
<>	Vérifier que la partie à gauche soit différente de la partie à droite du signe Exemple: WHERE price <> 20

Description

Operateur	Description
AND	Ajouter plusieurs conditions
BETWEEN	Rechercher des valeurs entre deux limites
IN	Rechercher des valeurs dans un ensemble défini
NOT	Negation de la condition
OR	Ajouter plusieurs conditions
LIKE	Trouver les observations similaires à une valeur choisie
NULL	Vérifier la présence de valeurs manquantes

Utilisation de LIKE pour des caractères

Avec **LIKE**, la condition est plus vaste que le “=”:

WHERE status LIKE "clos%" Selection des status commençant par "clos"
WHERE status LIKE "%ed"
WHERE status LIKE "%ose%"

Exemple :

```
SELECT station_id as identifiant, name as nom  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
WHERE status LIKE "clos%"  
LIMIT 10
```

Exercices

Requête avec les filtres sur le WHERE

Questions

Sélectionner:

1. La table bikeshare_trips et les colonnes trip_id, bikeid
2. Ajouter à la question 1 les trajets partant de la station 2502
3. Toutes les colonnes et les trajets de plus de 30 min
4. Les trajets partant de la station 2502 et de plus de 25 min
5. Les trajets partant de la station 2502 ou de plus de 25 min
6. Les trajets entre 0 et 10 minutes ou entre 30 et 60 minutes
7. Les trajets en 2021
8. Les trajets en mars 2021

Utilisation des TIMESTAMP

Les opérations sur les dates peuvent très vite devenir compliqué si le format utilisé n'est pas le bon.



Utilisation des DATES

DATE transformer en format date

DATE_ADD ajouter du temps à une date

DATE_SUB enlever du temps à une date

DATE_DIF connaître la différence de temps qui sépare 2 dates

Exemple :

```
SELECT *, DATE_ADD( start_time, INTERVAL duration_minutes MINUTE)
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`
WHERE start_station_name="10th & Red River"
    AND end_station_name="Henderson & 9th"
```

Utilisation du TEMPS

Les même fonctions peuvent être utilisées avec le temps

Exemple :

```
SELECT TIME_ADD(TIME( start_time), INTERVAL duration_minutes MINUTE)
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`
WHERE start_station_name="10th & Red River"
      AND end_station_name="Henderson & 9th"
```

Exercices

Requête avec des fonctions liées au temps/durée

Questions

Sélectionner:

1. La table bikeshare_trips et les trajets le samedi
2. Les trajets en 2021
3. Les trajets en mars 2021

Ajout de colonnes

Ajouter une nouvelle colonne dans SELECT



Opérations

Ajout de nouvelles colonnes via une opération sur des formats numériques:
addition soustraction
multiplication division

Opérations

Ajout de nouvelles colonnes via une opération sur des formats numériques:

addition (scalaire + attribut) soustraction
multiplication division

Exemple :

```
SELECT 5 + duration_minute as end_time,  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
LIMIT 10
```

Opérations

Ajout de nouvelles colonnes via une opération sur des formats numériques:

addition (attribut + attribut)	soustraction
multiplication	division

Exemple :

```
SELECT fare+tips+tolls+extras as total_price  
FROM `bigquery-public-data.chicago_taxi_trips.taxi_trips`  
LIMIT 10
```

Opérations

Ajout de nouvelles colonnes via une opération sur des formats numériques:

addition

soustraction

multiplication

division

Exemple :

```
SELECT 30 - duration_minute as temps_restant_gratuit,  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
LIMIT 10
```

Opérations

Exemple :

```
SELECT 60*duration_minute as duration_seconde,  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
LIMIT 10
```

Opérations

Ajout de nouvelles colonnes via une opération sur des formats numériques:

Exemple :

```
SELECT duration_minute/60 as duration_heure,  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
LIMIT 10
```

Concaténation

Ajout de nouvelles colonnes via une addition de colonnes sous format STRING

Exemple :

```
SELECT CONCAT(hitterLastName, " ", hitterFirstName) as hitterName  
FROM `bigquery-public-data.baseball.games_post_wide`  
LIMIT 10
```

Fonctionnement de la Concaténation

`CONCAT(hitterLastName, " ", hitterFirstName)`

revient à lire le premier élément, puis l'élément après la virgule, puis l'élément suivant.

Exemple :

```
SELECT CONCAT(hitterLastName, " ", hitterFirstName) as hitterName  
FROM `bigquery-public-data.baseball.games_post_wide`  
LIMIT 10
```

`CONCAT(ROSAMONT, " ", Kevin) —> "ROSAMONT Kevin"`

CASE operator

Ajout de nouvelles colonnes prenant des valeurs conditionnelles:

Syntaxe :

```
CASE WHEN cond 1 THEN valeur 1  
      WHEN cond 2 THEN valeur 2
```

...

```
      WHEN cond N THEN valeur N  
      ELSE autre valeur
```

```
END as Name colonne
```

CASE operator

Ajout de nouvelles colonnes prenant des valeurs conditionnelles:

Exemple :

```
SELECT *,  
      CASE WHEN number_of_docks < 10 THEN "petite"  
            WHEN number_of_docks < 20 THEN "moyenne"  
            ELSE "grande"  
      END AS number_of_docks  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
LIMIT 10
```

CASE operator (meilleur exemple)

Ajout de nouvelles colonnes prenant des valeurs conditionnelles:

Exemple :

```
SELECT *,  
       CASE WHEN number_of_docks < 10 THEN "petite"  
             WHEN number_of_docks < 20 THEN "moyenne"  
             WHEN number_of_docks >= 20 THEN "grande"  
             ELSE NULL  
       END AS number_of_docks  
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`  
LIMIT 10
```

Fonctions d'agrégation

Comme sur Excel, des fonctions déjà existantes permettent la creation de nouvelles colonnes MIN(), MAX(), AVG(), SUM(), COUNT()

Exemple :

```
SELECT COUNT(DISTINCT station_id) as nb_open_station
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations`
WHERE status <> "closed"
LIMIT 10
```

Exercices

Requête avec des fonctions ajoutant des colonnes calculées

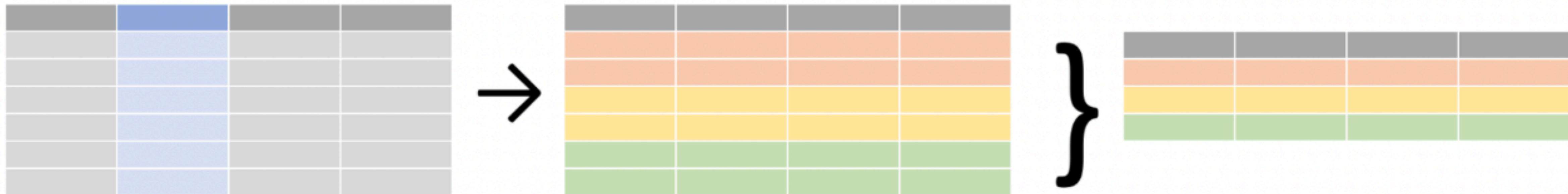
Questions

Sélectionner:

1. Dans la table `chicago_taxi_trips.taxi_trips`, quel pourcentage du montant total représente la course (fare)?
2. Dans la même table, prenez uniquement les voyages où le temps et les miles sont supérieurs à 0 puis convertir les miles en km. Créer 2 colonnes, le prix de la course par seconde et le prix de la course par km?
3. La table `austin_bikeshare.bikeshare_trips`, quel est le trajet le plus long en 2021.
4. La table `austin_bikeshare.bikeshare_trips`, quel est le plus vieux trajet enregistré?
5. La table `austin_bikeshare.bikeshare_trips`, en moyenne un trajet dure combien de temps?

Opération par groupe

GROUP BY



Fonctions d'agrégation

Les colonnes initiales dans le **GROUP BY** doivent apparaître dans le **SELECT**

Exemple :

```
SELECT start_station_id, end_station_id,  
       COUNT(DISTINCT trip_id) as nb_trajets  
  FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
 GROUP BY start_station_id, end_station_id  
 LIMIT 10
```

Fonctions d'agrégation

Les colonnes initiales dans le **GROUP BY** doivent apparaître dans le **SELECT**
(Comme c'est le cas de **start_station_id** et **end_station_id**)

Exemple :

```
SELECT start_station_id, end_station_id,  
       COUNT(DISTINCT trip_id) as nb_trajets  
  FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
 GROUP BY start_station_id, end_station_id  
 LIMIT 10
```

Exercices

Requête avec des opérations de groupe

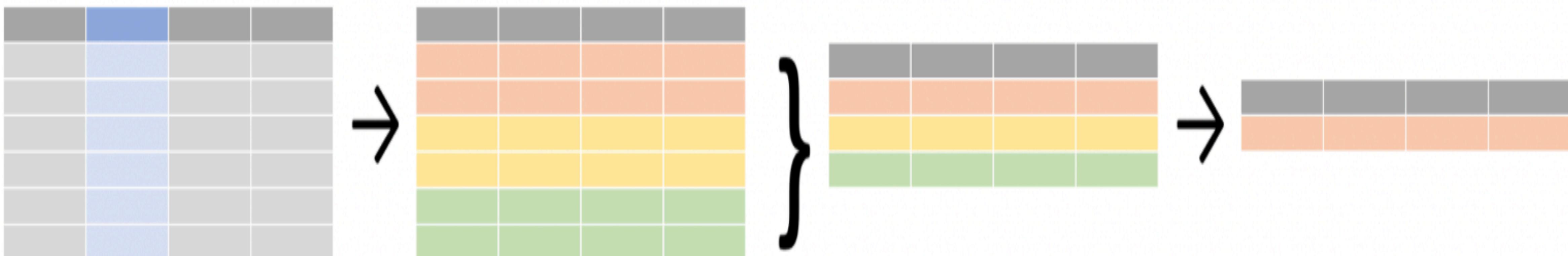
Questions

Sélectionner:

1. La table `baseball.games_post_wide`, donner la plus forte
attendance par équipe à domicile sur un match?
2. La table `baseball.games_post_wide`, donner les
attendance moyenne par équipe
3. La table `austin_bikeshare.bikeshare_trips`, la durée
moyenne du trajet par usager

Filtrer les opération par groupe

HAVING



Filtre sur les nouvelles colonnes

Les colonnes initiales peuvent être filtrées avec **WHERE**
mais les agrégats (ou colonnes calculées) doivent être filtrer avec **HAVING**

Exemple :

```
SELECT start_station_name, end_station_name,  
       COUNT(trip_id) as nb_trajets  
  FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips`  
 GROUP BY start_station_id, end_station_id  
 HAVING COUNT(trip_id) > 20  
 LIMIT 10
```

Pourquoi ne pas utiliser le WHERE?

(comme pour les colonnes déjà existantes)

SQL est un langage déclaratif
Le code ne s'execute pas dans l'ordre dans lequel il apparaît:

FROM
WHERE
GROUP BY
HAVING
SELECT
DISTINCT
ORDER BY
LIMIT

Requête type

```
SELECT DISTINCT column1 [, column2, AGG_FUNC(column_or_expression), ...]
FROM mytable
WHERE constraint_expression1 [AND/OR constraint_expression2...]
GROUP BY column1 [, column2,...]
HAVING constraint_having_expression1 [AND/OR constraint_having_expression2...]
ORDER BY column1 ASC/DESC [,column2,...]
LIMIT count;
```

LES JOINTURES

Travailler avec plusieurs tables

Associer plusieurs tables

Certaines questions nécessitent l'usage de plusieurs tables

D'où la nécessité de faire des schémas relationnels

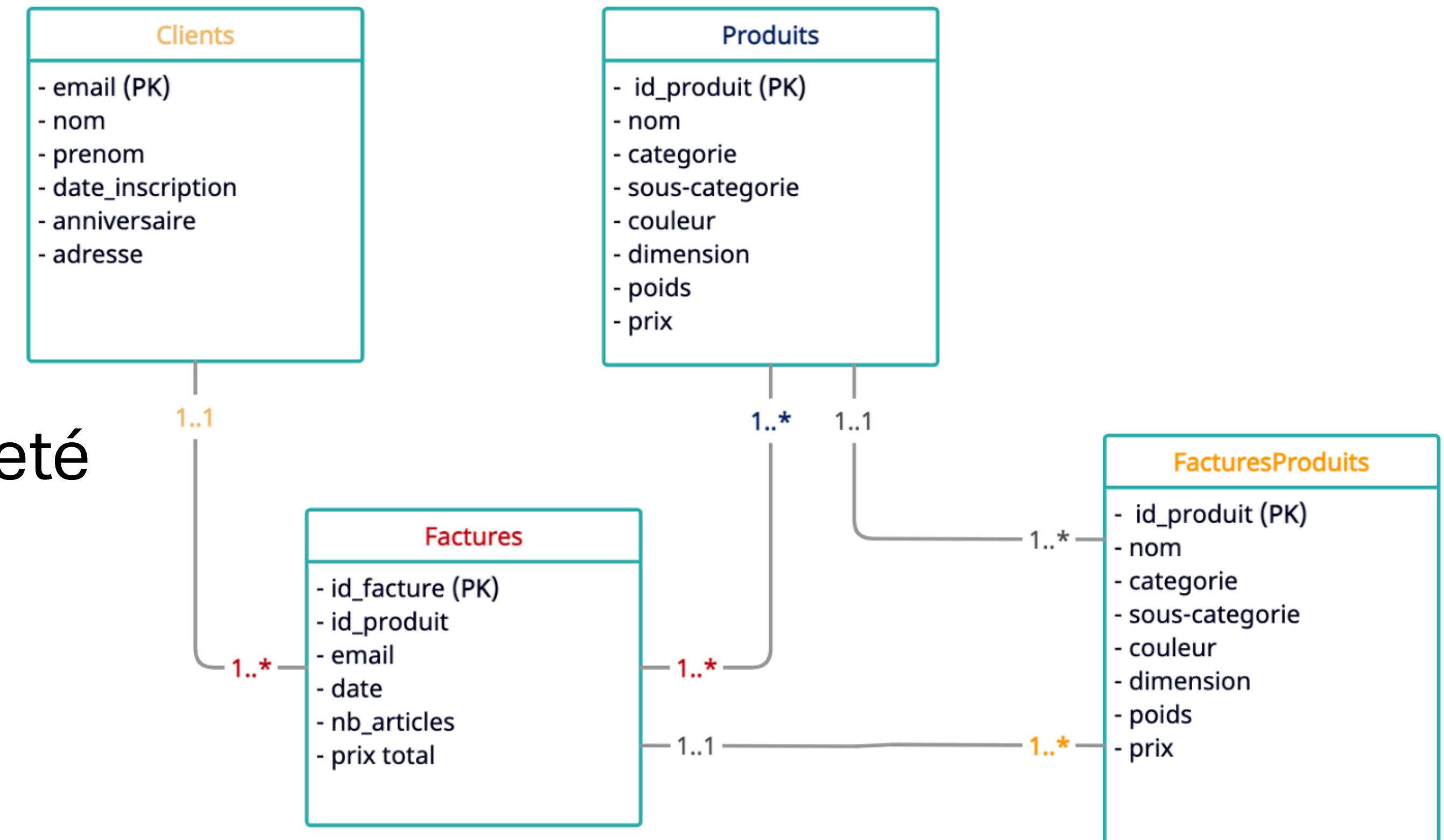
Revenons à l'exemple d'Amazon

Associer plusieurs tables

Si je veux connaître :

Le nom du produit le plus acheté

Le prénom du plus gros client



Retour aux notions de clefs

Table des trajets

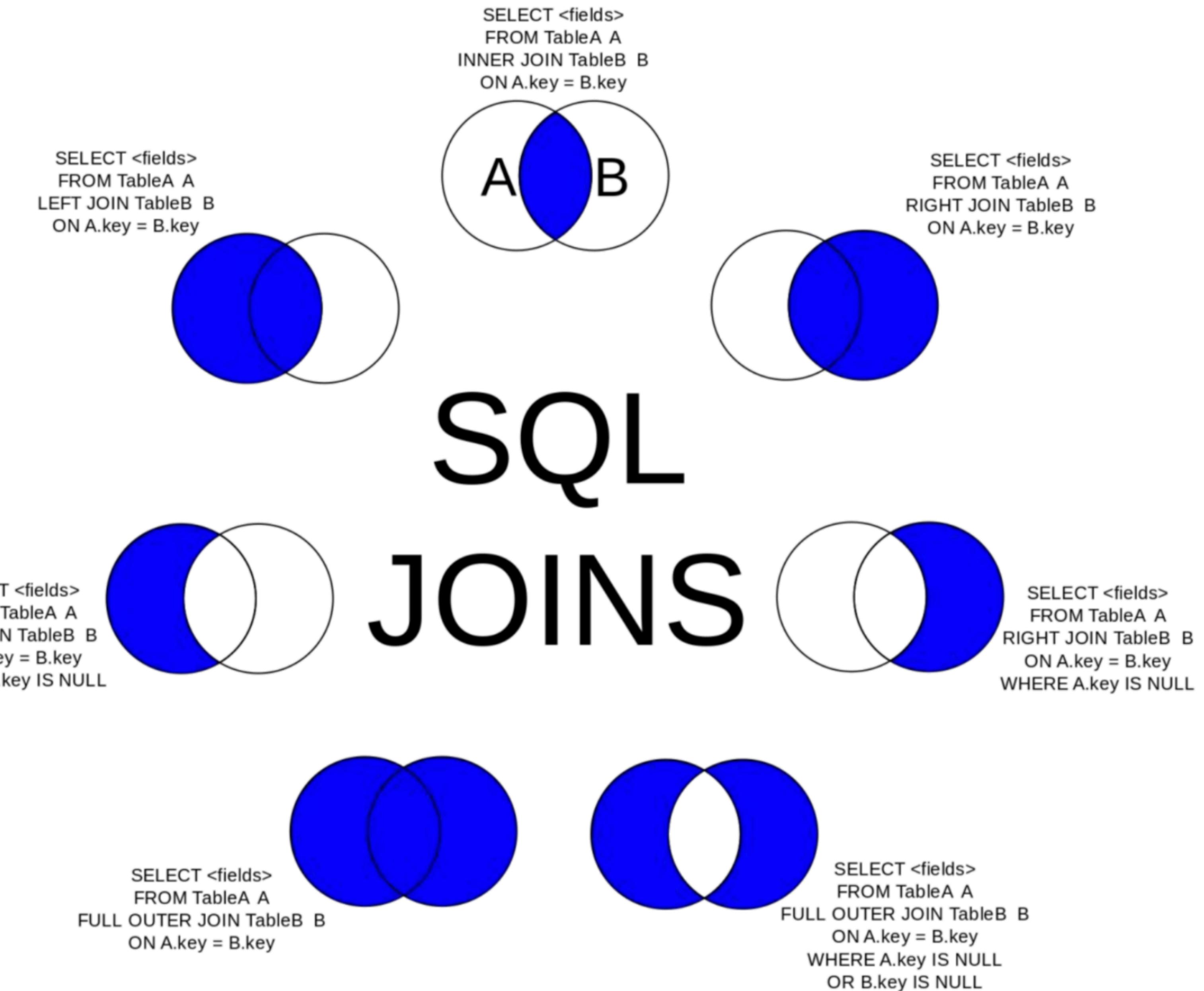
station_id	name	status	address
3464	Pease Park	closed	1155 Kingsbury St
2500	Republic Square	closed	425 W 4th Street
2536	Waller & 6th St.	closed	602 Waller St.
2538	Bullock Museum @ Congress & MLK	closed	1881 Congress Ave.
2541	State Capitol @ 14th & Colorado	closed	206 W. 14th St.
2545	ACC - Rio Grande & 12th	closed	700 W. 12th St.
2546	ACC - West & 12th Street	closed	1231 West Ave.
2550	Republic Square @ Guadalupe & 4th St.	closed	Presented by Austin Ventures

Retour aux notions de clefs

Table des trajets

trip_id	subscriber_type	bikeid	start_time	start_station_id	start_station_name	end_station_id
9900289692	Walk Up	248	2015-10-02 21:12:01 UTC	1006	Zilker Park West	1008
9900285987	24-Hour Kiosk (Austin B-cycle)	446	2014-10-26 15:12:00 UTC	2712	Toomey Rd @ South Lamar	2712
9900285989	24-Hour Kiosk (Austin B-cycle)	203	2014-10-26 15:12:00 UTC	2712	Toomey Rd @ South Lamar	2712
9900285991	24-Hour Kiosk (Austin B-cycle)	101	2014-10-26 15:12:00 UTC	2712	Toomey Rd @ South Lamar	2712
9900286140	24-Hour Kiosk (Austin B-cycle)	242	2014-10-26 18:12:00 UTC	2541	State Capitol @ 14th & Colorado	2541
9900286143	24-Hour Kiosk (Austin B-cycle)	924	2014-10-26 18:12:00 UTC	2541	State Capitol @ 14th & Colorado	2541
9900286171	24-Hour Kiosk (Austin B-cycle)	869	2014-10-26 18:12:00 UTC	2536	Waller & 6th St.	2536
9900286214	Annual Membership (Austin B-cycle)	24	2014-10-26 20:12:00 UTC	2712	Toomey Rd @ South Lamar	2712
9900286540	24-Hour Kiosk (Austin B-cycle)	117	2014-10-27 15:12:00 UTC	2536	Waller & 6th St.	2536

En SQL, les jointures
s'écrivent



SQL INNER JOIN

Velo

id_velo	marque	lot
1	A	20B
2	A	21A
4	C	23R

Station

id_stn	place	name
A	12	Bastille
B	15	Halles
D	8	Voltaire

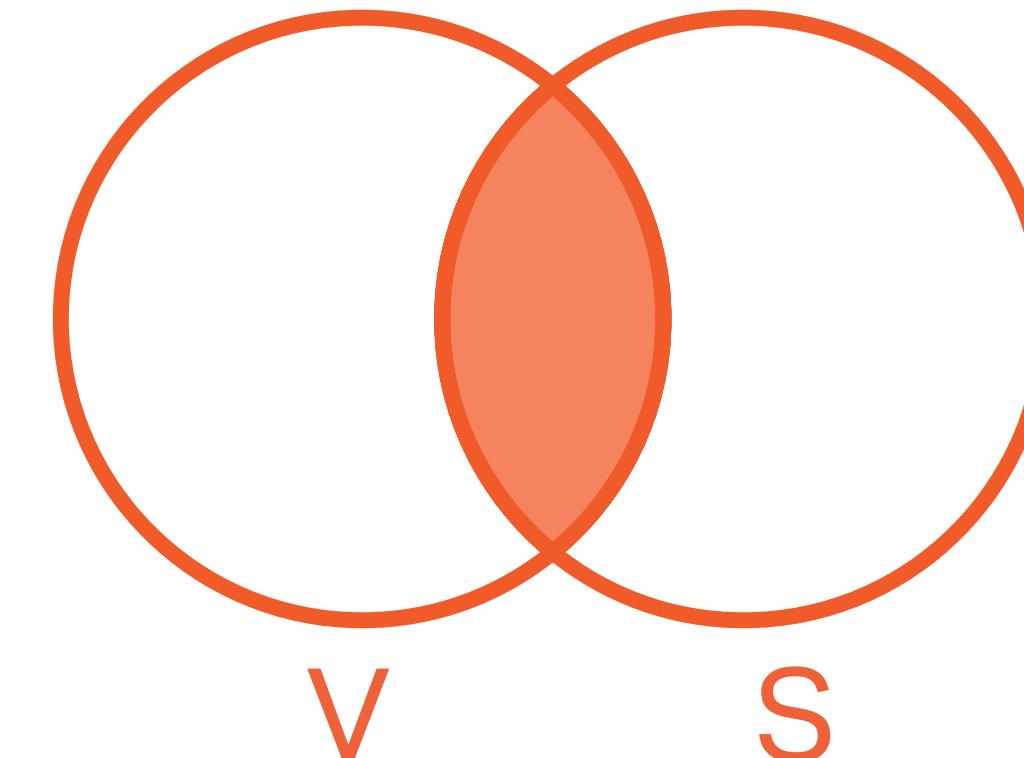
SQL INNER JOIN

Velo

id_velo	id_stn	lot
1	A	20B
2	A	21A
4	C	23R

Station

id_stn	place	name
A	12	Bastille
B	15	Halles
D	8	Voltaire



Intersection de **V** et **S**

Résultat

Exemple :

```
SELECT *
FROM Velo as V
INNER JOIN Station as S
ON S.id_stn = V.id_stn
```

id_velo	id_stn	name	place	lot
1	A	Bastille	12	20B
2	A	Bastille	12	21A

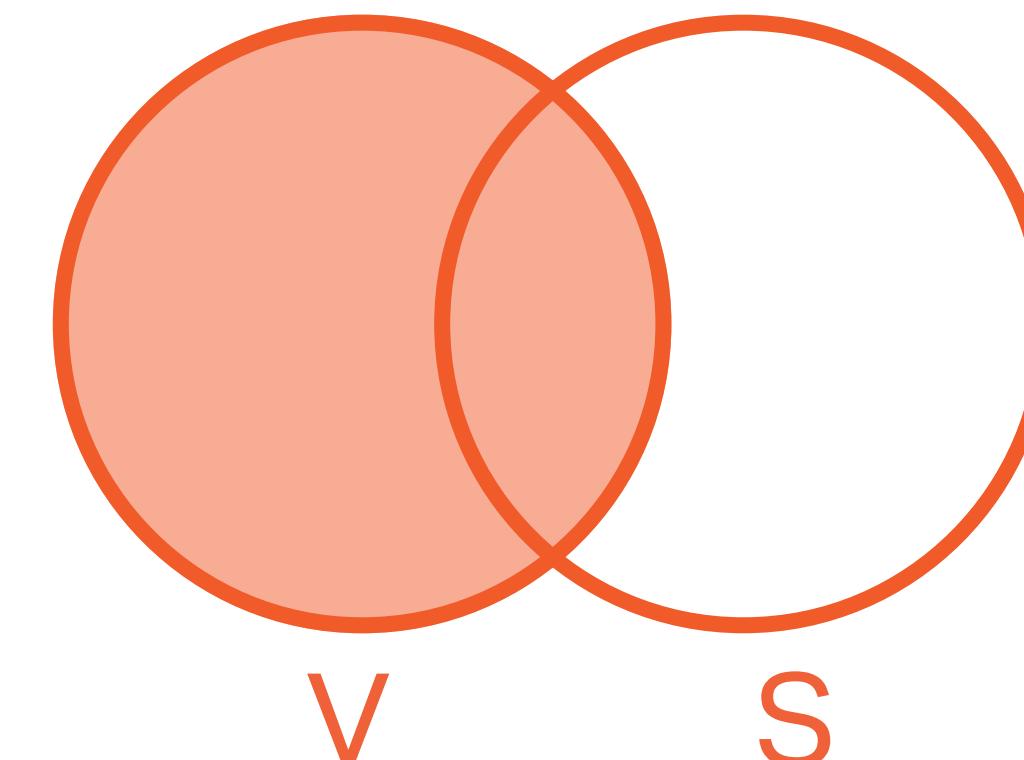
SQL LEFT JOIN

Velo

id_velo	id_stn	lot
1	A	20B
2	A	21A
4	C	23R

Station

id_stn	place	name
A	12	Bastille
B	15	Halles
D	8	Voltaire



Tous les éléments de **V** et les éléments de **S** qui se retrouve dans **V**

Résultat

Le nombre de ligne du résultat dépendra du nombre de ligne dans la table du FROM

Exemple :

```
SELECT *
FROM Velo as V
LEFT JOIN Station as S
ON S.id_stn = V.id_stn
```

id_velo	id_stn	name	place	lot
1	A	Bastille	12	20B
2	A	Halles	12	21A
4	C			23R

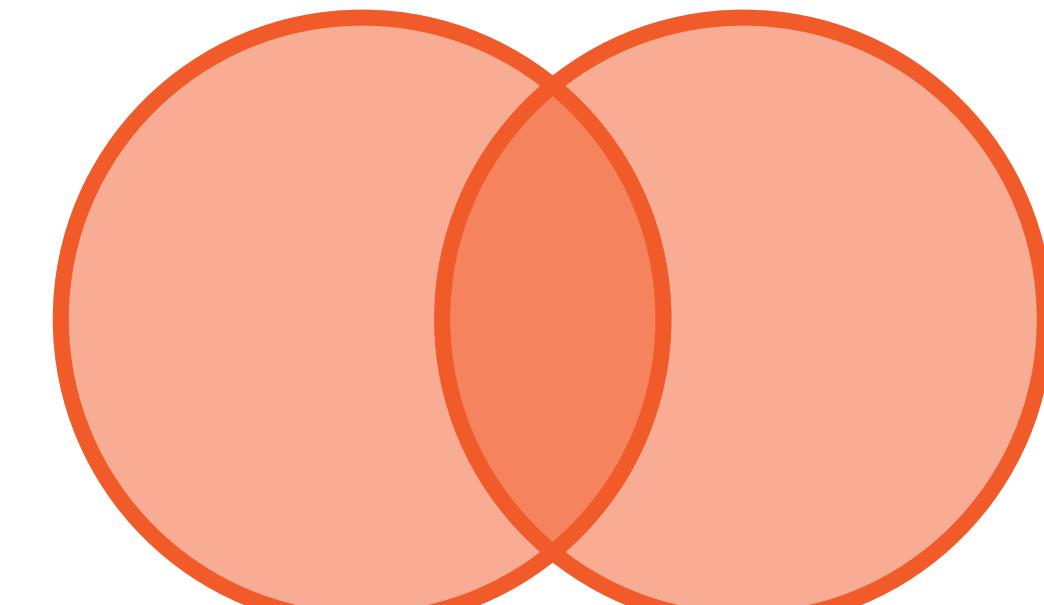
SQL FULL OUTER JOIN

Velo

id_velo	id_stn	lot
1	A	20B
2	A	21A
4	C	23R

Station

id_stn	place	name
A	12	Bastille
B	15	Halles
D	8	Voltaire



V S

Tous les éléments de
V et de S

Résultat

Exemple :

```
SELECT *
FROM Velo as V
FULL OUTER JOIN Station as S
ON S.id_stn = V.id_stn
```

id_velo	id_stn	name	place	lot
1	A	Bastille	12	20B
2	A	Halles	12	21A
4	C			23R
	B	Halles	15	
	D	Voltaire	8	

Syntaxes des jointures

Si l'on veut récupérer l'intersection des 2 tables

Exemple :

```
SELECT trip_id, subscriber_type, start_station_name, address, status
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations` as s
INNER JOIN `bigquery-public-data.austin_bikeshare.bikeshare_trips` as t
ON s.station_id = t.start_station_id -- clef primaire à clef étrangère
LIMIT 20
```

Syntaxes des jointures

Si l'on veut récupérer l'ensemble des stations, même celle où il n'y a pas de trajets

Exemple :

```
SELECT trip_id, subscriber_type, start_station_name, address, status
FROM `bigquery-public-data.austin_bikeshare.bikeshare_stations` as s
LEFT JOIN `bigquery-public-data.austin_bikeshare.bikeshare_trips` as t
ON s.station_id = t.start_station_id -- clef primaire à clef étrangère
LIMIT 20
```

Syntaxes des jointures

Vous pouvez joindre autant de table que vous voulez

Exemple :

```
SELECT *
FROM `table1` as t1
LEFT JOIN `table2` as t2
ON t1.key1 = t2.key1 -- clef primaire à clef étrangère
LEFT JOIN `table3` as t3
ON t1.key2 = t3.key2      -- option si 3 eme table
LIMIT 20
```

Syntaxes des jointures

Vous pouvez joindre autant de table que vous voulez

Exemple :

```
SELECT * [t1.nom as t_nom] ←  
FROM `table1` as t1  
LEFT JOIN `table2` as t2  
ON t1.key1 = t2.key1 -- clef primaire à clef étrangère  
LEFT JOIN `table3` as t3  
ON t1.key2 = t3.key2  
LIMIT 20
```

Essayez t1.* ou t2.* à la place de *

(vous sélectionnerez les colonnes de t1 ou t2)

Exercices

Questions

1. Ouvrir la table `top_rising_terms` dans le dataset `google_trends` et me dire à quoi correspond les différentes colonnes
2. Ouvrir la table `top_terms` dans le dataset `google_trends` et me dire à quoi correspond les différentes colonnes
3. Afficher les top 5 des `top_terms` pour la semaine `2022-01-30` par lieu
4. Afficher les top 5 des `top_rising_terms` pour la semaine `2022-01-30` par lieu
5. Afficher le top 5 des `top_terms` et `top_rising_terms` pour la semaine `2022-01-30` par lieu

Ressources pour aller plus loin

Formation en ligne

<https://www.codecademy.com/learn/learn-sql>

<https://www.udacity.com/course/intro-to-relational-databases--ud197>

Livre

<http://philip.greenspun.com/sql/>

http://www.informit.com/library/library.aspx?b=STY_Sql_24hours



linkedin.com/in/krosamont/

MERCI

Contactez moi par message
si vous avez des questions

Kevin Rosamont