

Base de données

Cours 4 - Opérations sur les données stockées (CRUD)

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Les couches d'une base de données**
- 2 Interrogation sur les données**
 - Les bases
 - Avancées
 - Tris et Limites
- 3 Principe des Actions Destructrices**
- 4 Insertion**
 - Singulière
 - Multiple
- 5 Modification**
 - Singulière
 - Multiple
- 6 Suppression**
 - Singulière
 - Multiple

Les couches d'une base de données

Une base de données contient différentes composantes n'ayant pas le même niveau de malléabilité.

Les opérations sur les données **sont le comportement normal d'utilisation dans une situation de production.**

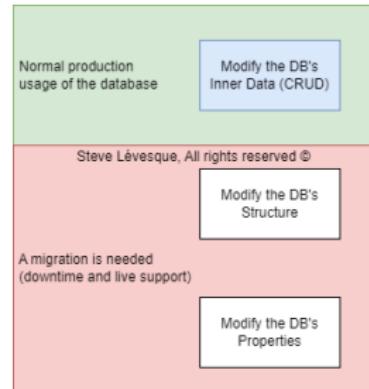


Figure: Les composantes d'une base de données

Interrogation sur les données

Il est indispensable pour une base de données d'offrir une logique pour lire, écrire, mettre à jour, ou supprimer les données contenues dans celle-ci.

Les interrogations, ou dans d'autres mots, la lecture sur les données, permettent d'extraire de l'information d'une base de données.

Utile lors de la **prise de décision** dans un contexte d'affaire.

Interrogation sur les données

On considère toutes les opérations précédentes comme le CRUD :

- Create (Créer)
- Read (Lire)
- Update (Mettre à jour)
- Delete (Supprimer)

Interrogation sur les données

Les 4 types d'interrogations suivantes seront abordés :

- Les bases (lecture directe)
- Avancées (avec contraintes arbitraires)
- Tris et Limites

Interrogation sur les données - Les bases

Il est possible d'obtenir **tous les documents** d'une base de données avec **SELECT**.

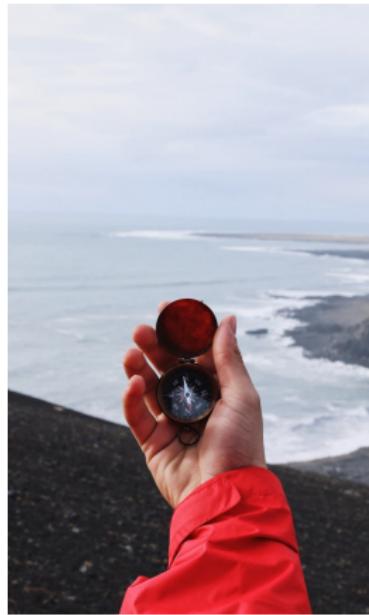


Figure: Opérande "SELECT" du langage SQL

Interrogation sur les données - Les bases

Listing: <https://www.postgresql.org/docs/current/sql-select.html>

```
1 SELECT * FROM course;
```

Interrogation sur les données - Avancées



Figure: Trouver des valeurs spécifiques avec SELECT

Il est possible d'obtenir **une étendue d'entrées (lignes/rangées)** d'une base de données avec des conditions dans le SELECT.

Par exemple :

- WHERE
- SELECT [Column1, Column2, ..., ColumnN] FROM
- Etc.

Interrogation sur les données - Avancées

Listing: <https://www.postgresql.org/docs/current/sql-select.html>

```
1 SELECT course_name FROM course WHERE course_id > 110;
```

Interrogation sur les données - Tris et Limites

Il est possible d'obtenir **une étendue triée et limitée** d'une base de données avec **ORDER BY** et **LIMIT**.

La partie de triage ("ORDER BY") prend le nom de la colonne ainsi que la dénotation ascendante ("ASC") ou descendante ("DESC").

La partie de limitation ("LIMIT") permet de réduire le résultat final au nombre spécifique en paramètre. Si le nombre de documents est inférieur, cette quantité est retournée par défaut.



Figure:
`collection.find(query).sort(rule).limit(number)`

Interrogation sur les données - Tris et Limites

Listing: <https://www.postgresql.org/docs/current/sql-select.html>

```
1 SELECT course_name FROM course WHERE course_id > 110  
2 ORDER BY course_name ASC LIMIT 5;
```

Principe des Actions Destructrices

Une action est considérée **destructrice** lorsqu'elle menace l'intégrité de la base de données.

Dans cette définition large, l'insertion est considérée comme une menace à l'intégrité de la base de données, et donc une action destructrice sur son intégrité.

L'interrogation serait la seule manipulation préservant l'état d'une base de données.

Insertion

L'insertion permet d'ajouter de nouveaux objets (entrées) dans la base de données et d'augmenter le corpus de la base de données.

Insertion - Singulière

Une insertion singulière permet de sauvegarder une entrée (au respect des contraintes de validation des tables) dans la base de données.

La commande est : **INSERT INTO [Table name] (Column1, ..., ColumnN) Values (...)**, où l'entrée est un seul objet.



Figure: **INSERT INTO**

Insertion - Singulière

Listing: <https://www.postgresql.org/docs/current/sql-insert.html>

```
1 INSERT INTO course (course_id, course_code, course_name, department, instructor)
2 VALUES (121, 'IOT121', 'Internet of Things', 'Computer Science', 'Dr. Silver');
```

Insertion - Multiple

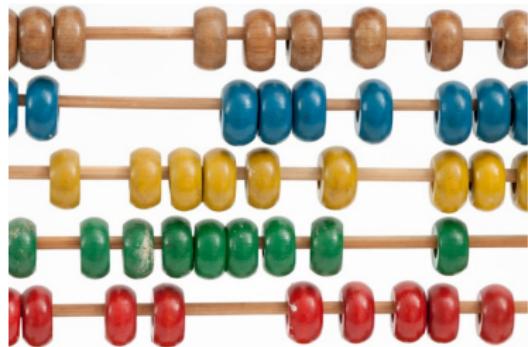


Figure: **INSERT INTO**

Une insertion multiple permet de sauvegarder plusieurs entrées d'un seul coup dans la base de données.

La commande est la même et il suffit simplement d'enchaîner plusieurs entrées une après l'autre.

Insertion - Multiple

Listing: <https://www.postgresql.org/docs/current/sql-insert.html>

```
1 INSERT INTO course (course_id, course_code, course_name, department, instructor
2   (122, 'SEC122', 'Cyber Security Essentials', 'Computer Science', 'Ms. Violet
3   (123, 'DSA123', 'Data Structures & Algorithms', 'Computer Science', 'Dr. Inc
4   (124, 'CLD124', 'Cloud Computing Fundamentals', 'Information Technology', 'I
5   (125, 'ML125', 'Machine Learning Introduction', 'Computer Science', 'Dr. Rub
```

Modification

La modification permet de mettre à jour des documents (entrées) qui peuvent être d'une représentation fautive ou non optimale.

Par exemple, il est important en Intelligence Artificielle que les données soient **propres, normalisées (entre 0 et 1) et absentes de valeurs considérées comme des vides.**

Modification - Singulière

Une modification singulière permet de changer une ou plusieurs valeurs (colonnes) d'un objet (entrée) spécifique dans la base de données.

La commande est : **UPDATE [Table name] SET [Column name] WHERE [Column name] [Math. comparison]**, où l'entrée est un seul objet.



Figure: **UPDATE [Table name] SET [Column name] WHERE [Column name] [Condition]**

Modification - Singulière

Listing: <https://www.postgresql.org/docs/current/sql-update.html>

```
1 UPDATE course
2 SET instructor = 'Prof. Levesque'
3 WHERE course_id = 101;
```

Modification - Multiple



Figure: **UPDATE [Table name] SET [Column name]**

Une modification multiple permet de changer des valeurs de plusieurs entrées respectant les contraintes de recherche, le tout d'un seul coup dans la base de données.

La commande globale est :
UPDATE [Table name] SET [Column name], et celle avec une délimitation est accompagnée d'un "WHERE" permissif sur une étendue.

Modification - Multiple

Listing: <https://www.postgresql.org/docs/current/sql-update.html>

```
1 UPDATE course  
2 SET instructor = 'Prof. Levesque';
```

Modification - Multiple

Listing: <https://www.postgresql.org/docs/current/sql-update.html>

```
1 UPDATE course
2 SET instructor = 'Prof. ABC'
3 WHERE course_id IN (101, 102, 103);
```

Suppression

La suppression permet d'enlever des entrées (lignes) incohérents, inutiles et nuisant au corpus et à l'intégrité des données.

Par exemple, si une classe dans un problème de classification en Intelligence Artificielle n'est plus utile pour le domaine d'affaires et que celle-ci fausse les résultats, il serait pertinent de la supprimer de la base de données.

Même principe si des données peuvent impacter la prise de décision de manière positive, dans le cas où elle sont enlevées.

Suppression - Singulière

Une suppression singulière permet d'enlever un document spécifique dans la base de données.

La commande est : **DELETE FROM [Table name] WHERE [Condition].**

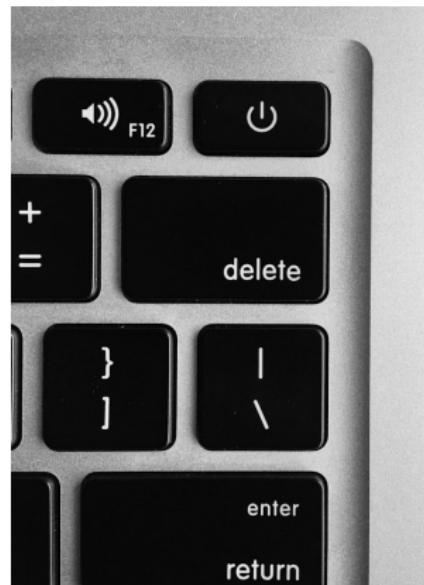


Figure: **DELETE FROM [Table name] WHERE [Condition]**

Suppression - Singulière

Listing: <https://www.postgresql.org/docs/current/sql-delete.html>

```
1 DELETE FROM course  
2 WHERE course_id = 101;
```

Suppression - Multiple



Une suppression multiple permet d'enlever plusieurs entrées respectant les contraintes de recherche, le tout d'un seul coup dans la base de données.

La commande est : **DELETE FROM [Table name] WHERE 1=1 ou TRUNCATE TABLE [Table name].**

Figure: **DELETE FROM [Table name] WHERE 1=1 ou TRUNCATE TABLE [Table name]**

Suppression - Multiple

Listing: <https://www.postgresql.org/docs/current/sql-delete.html>

```
1 -- This works, but not in a readable/optimal way.  
2 DELETE FROM course WHERE 1= 1;  
3  
4 -- The official one-liner to remove all rows from a table.  
5 -- It provides a faster mechanism to remove rows.  
6 TRUNCATE TABLE course;
```

Suppression - Multiple

Listing: <https://www.postgresql.org/docs/current/sql-delete.html>

```
1 DELETE FROM course  
2 WHERE course_id < 111;
```

Bibliographie

- <https://www.postgresql.org/docs/current/sql-select.html>
- <https://www.postgresql.org/docs/current/sql-insert.html>
- <https://www.postgresql.org/docs/current/sql-update.html>
- <https://www.postgresql.org/docs/current/sql-delete.html>