

# TP1 - BASES DE DONNEES

## I. DECOUVERTE DE SQLite

---

Grâce à cette première partie nous avons pu découvrir les commandes nécessaires à la création d'une base de données et de tables pour la composer. Nous avons aussi pu remplir ses tables pour tester des requêtes simples. Par exemple, lister les noms des étudiants qui suivent un cours.

## II. REQUETES SQL

---

Après avoir construit la base de données, nous avons effectué les requêtes SQL correspondant aux questions du TD1. Nous n'avons traité que les 6 premières requêtes en TD donc nous ne sommes pas allées beaucoup plus loin en TP.

Nous avons minutieusement comparé nos résultats avec les différentes tables de la base de données pour être certaines que ceux-ci ne sont pas erronés.

## III. QUERY PLAN

---

La formulation française de la requête proposée est : « Trouver le nom du fournisseur et le nom du produit pour chacune des fournitures ».

Le query plan correspond aux étapes par lesquelles le système est passé afin de renvoyer le résultat d'une requête.

Le query plan de la requête proposée est :

QUERY PLAN

|--SCAN fs

|--SEARCH ft USING AUTOMATIC COVERING INDEX (f=?)

`--SEARCH p USING AUTOMATIC COVERING INDEX (p=?)

Cela correspond à :

- 1) Parcours de la table Fournisseur

- 2) Parcours des identifiants Fournisseur de la table Fourniture à l'aide d'un indice qui cherche et trie en même temps
- 3) Parcours des identifiants Produit de la table Produit à l'aide d'un indice qui cherche et trie en même temps

Analysons le query plan de la requête suivante :

```
SELECT DISTINCT nom_p FROM Produit p
JOIN Fourniture fo ON fo.p = p.p
JOIN Fournisseur f ON fo.f = f.f
WHERE qte < 5 AND nom_f = "Bornibus";
```

La commande explain query plan nous renvoie les lignes suivantes :

QUERY PLAN

```
--SCAN f
--SEARCH fo USING AUTOMATIC PARTIAL COVERING INDEX (f=?)
--SEARCH p USING AUTOMATIC COVERING INDEX (p=?)
`--USE TEMP B-TREE FOR DISTINCT
```

Cela correspond à :

- 1) Parcours de la table Fournisseur
- 2) Parcours des identifiants Fournisseur de la table Fourniture à l'aide d'un indice qui cherche et trie en même temps
- 3) Parcours des identifiants Produit de la table Produit à l'aide d'un indice qui cherche et trie en même temps
- 4) La dernière ligne indique l'utilisation d'une structure de donnée du type arbre binaire de recherche afin d'éviter la redondance

On remarque que les query plans de ces deux requêtes sont très similaires. C'est sans doute car elles sont de la même forme : des jointures ou des produits cartésiens avec une condition (un « WHERE »).

## IV. Conclusion

---

Ce premier TP nous aura permis de prendre en main les commandes de base de SQLite. Il nous a également permis de comprendre la logique de base de la création de requêtes ainsi que le fonctionnement du système pour répondre à une requête grâce au query plan.