MySQL (Plan d'Exécution /Hint)

Etapes :

- Créer le schéma de base de données SSB
- Charger les fichiers de données
- Créer les indexes
- Exécuter les requêtes
- Visualiser le plan d'éxécution
- Utiliser les HINT pour :
  - Utiliser et ignorer les index
  - Type d'optimisation
  - Variable serveur

La commande SHOW : (MetaData)

- SHOW INDEX FROM NomTable\G
- SHOW CREATE TABLE
- SHOW INDEXES
- SOHOW TABLE STATUS
- INFORMATION_SCHEMA
- SHOW VARIALES

Création des indexes

- CREATE INDEX idx_Title ON items (price);
- CREATE BITMAP INDEX ON ingredients (unit);
- DROP INDEX idx_Title;

Function-Based Index Examples

SELECT first_name, last_name

FROM employees WHERE UPPER(last_name) IS NOT NULL

ORDER BY UPPER(last_name);

CREATE INDEX upper_ix ON employees (UPPER(last_name));

Influencing MySQL with Hints

Ignorer un index

```
select sum(lo_extendedprice*lo_discount) as revenue
from lineorder IGNORE INDEX
(LO_DISCOUNT_IDX),dates
where lo_orderdate=d_datekey and d_year='1993'
and lo_discount in (1,2,3) and lo_quantity=25;
```

Use index

```
SELECT * FROM mytable USE INDEX (mod_time, name
```

L'outil EXPLAIN

L'outil EXPLAIN donne le plan d'exécution d'une requête. La description comprend :

- Le chemin d'accès utilisé.
- Les opérations physiques (tri, fusion, intersection,...).
- L'ordre des opérations.
- Il est représentable par un arbre

```
• EXPLAIN <Requête>
```

HINT

```
SELECT /*! Write HINT */ columns
FROM table.
```

- **SQL_NO_CACHE :**Cas des requetes trop dynamique
- **SQL_CACHE :** Réglage : set query_cache_type = 2
  - **Vous pouvez utiliser le HINT « SQL_CACHE »**
    - SET GLOBAL query_cache_type=0;
    - SET GLOBAL query_cache_type=1;
    - SET GLOBAL query_cache_type=2;
  - **HIGH_PRIORITY** : Pour les requêtes de type SELECT et INSERT
  - **LOW_PRIORITY:** Pour les requêtes de type INSERT et UPDATE
  - **INSERT DELAYED :** Il reviendra immédiatement, mais il faudra encore attendre d'autres clients ont fermé la table avant d'exécuter l'instruction.
  - **STRAIGHT_JOIN :** Cette astuce va dire à MySQL pour joindre les tables dans l'ordre où ils sont spécifiés dans la clause FROM.
  - **SQL_BUFFER_RESULT :** Cet indicateur indique à MySQL de mettre le résultat de la requête dans une table temporaire. Cela permettra de libérer un verrou de table tandis que le jeu de résultats est envoyé au client. Donc vous ne voulez l'utiliser sur des jeux de résultats.
  - **SQL_BIG_RESULT :** SQL_BIG_RESULT peut être utilisé avec DISTINCT et GROUP BY SELECT. Comme vous pouvez le deviner, indique à MySQL que le jeu de résultats sera grand. Selon la documentation de MySQL, si elle est invoquée
- MySQL utilise directement les tables temporaires sur disque si nécessaire, et préfère le tri à l'aide d'une table temporaire avec une clé sur le GROUP BY éléments.

```
• SELECT HIGH_PRIORITY * FROM DimCustomer;
• SELECT SQL_NO_CACHE * FROM DimCustomer;
• SELECT SQL_BUFFER_RESULT * FROM counties;
• SELECT SQL_NO_CACHE COUNT(*) AS total,YEAR(birth_date) AS birth_year
  FROM counties;
```