

Bases de Données Relationnelles II

Optimisation et les index

L2

Rafael Angarita
Maitre de Conférences
rangarit@parisnanterre.fr

Optimisation

- Configuration, réglage et mesure des performances, à plusieurs niveaux.
- Optimisation des instructions SQL individuelles, des applications entières, d'un ou plusieurs serveurs de base de données.
- Vouvez être proactif ou vous pouvez dépanner une configuration ou un code.

Les performances

Plusieurs facteurs au niveau de la base de données, tels que les tables, les requêtes et les paramètres de configuration.

Les utilisateurs classiques cherchent à obtenir les meilleures performances de la base de données à partir de leurs configurations logicielles et matérielles existantes.

- Optimisation au niveau de la base de données
- Optimisation au niveau du matériel

Optimisation au niveau de la base de données

C'est le facteur le plus important !

- Les tables sont-elles structurées correctement ?
- Les bons index sont-ils en place ?
- Utilisez-vous le moteur de stockage approprié pour chaque table ?
- L'application utilise-t-elle une stratégie de verrouillage appropriée ?
- L'application utilise-t-elle une stratégie de verrouillage appropriée ?

Optimisation au niveau du matériel

C'est le facteur le plus important !

- Recherche sur le disque.
- Lecture et écriture sur le disque.
- Cycles du processeur.
- Bande passante de la mémoire.

Optimization de requêtes

- Utiliser des indices chaque fois que cela est approprié
 - Pour éviter de gaspiller de l'espace disque, créez un petit ensemble d'index qui accélèrent de nombreuses requêtes associées utilisées dans votre application.
 - Especially important for joins
 - Minimiser le nombre de balayages de table complets
- Evitez de complexifier les requêtes
 - L'optimiseur effectue automatiquement certaines des transformations
- Utilisez l'instruction EXPLAIN pour récupérer le plan d'exécution de la requête et ajustez vos index/requête
- Même les requêtes qui s'exécutent rapidement pourraient bénéficier de l'optimisation

Optimisations Automatiques

- MySQL optimise automatiquement vos requêtes, pour une exécution plus rapide
- Exemple d'optimisation de l'instruction WHERE :
 - Détecter les conditions qui ne s'appliquent jamais
 - Filtrer les valeurs inutiles dans chaque table avant JOIN
 - Les expressions constantes sont évaluées une fois

Exemple

- `SELECT year, Max(length)`
`FROM Movies`
`WHERE genre = "comedy"`
`GROUP BY year`
`HAVING Max(length) > 100`
- Comment pouvons-nous modifier cette requête pour améliorer les performances ?

Déplacez le prédicat de la clause `HAVING` vers la clause `WHERE`, ce qui entraîne moins de tuples dans chaque groupe

Example(2)

- `SELECT year, Max(length), count(*)
FROM Movies
WHERE genre = "comedy"
GROUP BY year
HAVING Max(length) > 100`
- L'optimisation que nous avons proposée serait-elle toujours valable pour la requête ci-dessus ?

Non, car nous devons compter toutes les comédies chaque année, y compris les courtes.

Les index

Index

Emmaus, Pennsylvania,
General Horticulture.

Mushrooms, National
Rick Warne, London

Globe Pequot Press:

Asplenium petiolate 38-40
Asplenium retroflexus 99-100
Asplenium laevis 89-91
Asplenium margaritacea 78-79
Asplenium Americana 42-44
Asplenium nudicaulis 45-46
Asplenium rusticana 50-51
Arrowhead (*Sagittaria*
latifolia) 87-88
Asplenium syriaca 57-60
Asplenium macrophyllum 30-31
Asplenium patula 115-117
Asplenium vulgare 36-38
Basket 3

Browne, William 14
Browne's Britannia's Pastorals 14
Browne's Pastorals 14
Bumblebee weed 67
Cakile edentula 114-115
Calvatica species 152-154
Canada Goldenrod (*Solidago*
canadensis) 127-128
Canned Fiddleheads 163
Cantharellus cibarius 140-142, 167
Cattail (*Typha latifolia*) 60-62
Cattail Sprouts 10-11
Chaga (*Inonotus obliquus*) 154-156
Chanterelles (*Cantharellus*

Les index

- Un index est une structure qui reprend la liste ordonnée des valeurs auxquelles il se rapporte.
- Les index sont utilisés pour accélérer les requêtes.
- Ils sont indispensables à la création de clés, étrangères et primaires, qui permettent de garantir l'intégrité des données.

Qu'est-ce qu'un index ?

Structure de données qui reprend la **liste ordonnée** des valeurs auxquelles il se rapporte.

—

Définition

Lorsque vous créez un index sur une table, MySQL stocke cet index sous forme d'une structure particulière, contenant les valeurs des colonnes impliquées dans l'index.

Cette structure stocke les valeurs triées et permet d'accéder à chacune de manière efficace et rapide.

Qu'est-ce qu'un index ?

Trouvez les noms pour les id ≤ 4

Id	Id	Espèce	Sexe	Date de naissance	Nom	Commentaires
1	2	chat	NULL	2010-03-24 02:23:00	Roucky	NULL
2	1	chien	male	2010-04-05 13:43:00	Rox	Mordille beaucoup
3	3	chat	femelle	2010-09-13 15:02:00	Schtroumpfette	NULL
4	6	tortue	femelle	2009-06-13 08:17:00	Bobosse	Carapace bizarre
5	9	tortue	NULL	2010-08-23 05:18:00	NULL	NULL
6	4	tortue	femelle	2009-08-03 05:12:00	NULL	NULL
7	7	chien	femelle	2008-12-06 05:18:00	Caroline	NULL
8	8	chat	male	2008-09-11 15:38:00	Bagherra	NULL
9	5	chat	NULL	2010-10-03 16:44:00	Choupi	Né sans oreille gauche

Combien de lignes doit on parcourir ?

Pourquoi ne pas trier la table, simplement ?

Qu'est-ce qu'un index ?

Id		Id	Espèce	Sexe	Date de naissance	Nom	Commentaires		Date de naissance
1	↗	2	chat	NULL	2010-03-24 02:23:00	Roucky	NULL	↖	2008-09-11 15:38:00
2	↘	1	chien	male	2010-04-05 13:43:00	Rox	Mordille beaucoup	↖	2008-12-06 05:18:00
3	→	3	chat	femelle	2010-09-13 15:02:00	Schtroumpfette	NULL	↖	2009-06-13 08:17:00
4	↗	6	tortue	femelle	2009-06-13 08:17:00	Bobosse	Carapace bizarre	↖	2009-08-03 05:12:00
5	↘	9	tortue	NULL	2010-08-23 05:18:00	NULL	NULL	↖	2010-03-24 02:23:00
6	↗	4	tortue	femelle	2009-08-03 05:12:00	NULL	NULL	↖	2010-04-05 13:43:00
7	↘	7	chien	femelle	2008-12-06 05:18:00	Caroline	NULL	↖	2010-08-23 05:18:00
8	↗	8	chat	male	2008-09-11 15:38:00	Bagherra	NULL	↖	2010-09-13 15:02:00
9	↘	5	chat	NULL	2010-10-03 16:44:00	Choupi	Né sans oreille gauche	↖	2010-10-03 16:44:00

Index sur id et date_naissance

Les index : quels avantages et désavantages ?

- Avantages :
 - Accélération des requêtes
 - Intégrité des données
- Désavantages :
 - Ils prennent de la place en mémoire
 - Ils ralentissent les requêtes d'insertion, modification et suppression, puisqu'à chaque fois, il faut remettre l'index à jour en plus de la table.


Index sur plusieurs colonnes

id	nom	prenom	init_2e_prenom	email
1	Dupont	Charles	T	charles.dupont@email.com
2	François	Damien	V	fdamien@email.com
3	Vandenbush	Guillaume	A	guillaumevdb@email.com
4	Dupont	Valérie	C	valdup@email.com
5	Dupont	Valérie	G	dupont.valerie@email.com
6	François	Martin	D	mdmartin@email.com
7	Caramou	Arthur	B	leroiarthur@email.com
8	Boulian	Gérard	M	gebou@email.com
9	Loupiot	Laura	F	loulau@email.com
10	Sunna	Christine	I	chrichrisun@email.com

si vous faites souvent des recherches sur les trois colonnes à la fois, il vaut encore mieux faire un seul index, sur les trois colonnes (*nom, prenom, init_2e_prenom*)

Index sur plusieurs colonnes

nom	prenom	init_2e_prenom		id	nom	prenom	init_2e_prenom	email
Boulian	Gérard	M		1	Dupont	Charles	T	charles.dupont@email.com
Caramou	Arthur	B		2	François	Damien	V	fdamien@email.com
Dupont	Charles	T		3	Vandenbush	Guillaume	A	guillaumevdb@email.com
Dupont	Valérie	C		4	Dupont	Valérie	C	valdup@email.com
Dupont	Valérie	G		5	Dupont	Valérie	G	dupont.valerie@email.com
François	Damien	V		6	François	Martin	D	mdmartin@email.com
François	Martin	D		7	Caramou	Arthur	B	leroiarthur@email.com
Loupiot	Laura	F		8	Boulian	Gérard	M	gebou@email.com
Sunna	Christine	I		9	Loupiot	Laura	F	loulau@email.com
Vandenbush	Guillaume	A		10	Sunna	Christine	I	chrichrisun@email.com



Index sur trois colonnes

Tirer parti des "index par la gauche"

Mais comment fait-on si l'on fait aussi souvent des recherches uniquement sur le nom, ou uniquement sur le prénom, ou sur le nom et le prénom en même temps mais sans l'initiale ?

Faut-il créer un nouvel index pour chaque type de recherche ?

Non !

Index sur plusieurs colonnes

prenom	nom	nom	prenom	nom	prenom	init_2e_prenom
Arthur	Boulian	Boulian	Gérard	Boulian	Gérard	M
Charles	Caramou	Caramou	Arthur	Caramou	Arthur	B
Christine	Dupont	Dupont	Charles	Dupont	Charles	T
Damien	Dupont	Dupont	Valérie	Dupont	Valérie	C
Gérard	Dupont	Dupont	Valérie	Dupont	Valérie	G
Guillaume	François	François	Damien	François	Damien	V
Laura	François	François	Martin	François	Martin	D
Martin	Loupiot	Loupiot	Laura	Loupiot	Laura	F
Valérie	Sunna	Sunna	Christine	Sunna	Christine	I
Valérie	Vandenbush	Vandenbush	Guillaume	Vandenbush	Guillaume	A

Index par la gauche

Les index en MySQL

Index dans la description de la colonne

```
1 CREATE TABLE nom_table (  
2     colonne1 INT KEY,           -- Crée un index simple sur colonne1  
3     colonne2 VARCHAR(40) UNIQUE, -- Crée un index unique sur colonne2  
4 );
```

Liste d'index

```
1 CREATE TABLE nom_table (  
2     colonne1 description_colonne1,  
3     [colonne2 description_colonne2,  
4     colonne3 description_colonne3,  
5     ...,]  
6     [PRIMARY KEY (colonne_clé_primaire)],  
7     [INDEX [nom_index] (colonne1_index [, colonne2_index, ...])  
8 )  
9 [ENGINE=moteur];
```

Liste d'index

```
1  CREATE TABLE Animal (  
2      id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
3      espece VARCHAR(40) NOT NULL,  
4      sexe CHAR(1),  
5      date_naissance DATETIME NOT NULL,  
6      nom VARCHAR(30),  
7      commentaires TEXT,  
8      PRIMARY KEY (id),  
9      INDEX ind_date_naissance (date_naissance),  -- index  
10     INDEX ind_nom (nom(10))                    -- index  
11 )  
12 ENGINE=INNODB;
```


Liste d'index

```
1  CREATE TABLE Animal (  
2      id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
3      espece VARCHAR(40) NOT NULL,  
4      sexe CHAR(1),  
5      date_naissance DATETIME NOT NULL,  
6      nom VARCHAR(30),  
7      commentaires TEXT,  
8      PRIMARY KEY (id),  
9      INDEX ind_date_naissance (date_naissance),  
10     INDEX ind_nom (nom(10)),  
11     UNIQUE INDEX ind_uni_nom_espece (nom, espece)  -- Index s  
12 )  
13 ENGINE=INNODB;
```

Ajout avec ALTER TABLE

```
1  ALTER TABLE nom_table
2  ADD INDEX [nom_index] (colonne_index [, colonne2_index ...]);
3
4  ALTER TABLE nom_table
5  ADD UNIQUE [nom_index] (colonne_index [, colonne2_index ...]);
```

```
1  ALTER TABLE Test_tuto
2  ADD INDEX ind_nom (nom);
```

Ajout avec CREATE INDEX

```
1  CREATE INDEX nom_index  
2  ON nom_table (colonne_index [, colonne2_index ...]);  
3  
4  CREATE UNIQUE INDEX nom_index  
5  ON nom_table (colonne_index [, colonne2_index ...]);  
6
```

```
1  CREATE INDEX ind_nom  
2  ON Test_tuto (nom);
```

Suppression d'un INDEX

```
1  ALTER TABLE nom_table  
2  DROP INDEX nom_index;
```

On ne peut pas modifier un index

Recherches avec FULLTEXT

Recherches avec FULLTEXT

```
CREATE FULLTEXT INDEX nom_index  
ON nom_table (colonne_index [, colonne2_index ...]);
```

Recherche naturelle

```
1 SELECT *  
2 FROM nom_table  
3 WHERE MATCH(colonne1[, colonne2, ...])  
4 AGAINST ('chaîne recherchée');
```

```
1 SELECT *  
2 FROM nom_table  
3 WHERE MATCH(colonne1[, colonne2, ...])  
4 AGAINST ('chaîne recherchée' IN NATURAL LANGUAGE MODE);
```

Recherche naturelle

```
1 SELECT *  
2 FROM Livre  
3 WHERE MATCH(auteur)  
4 AGAINST ('Terry');
```

id	auteur	titre
8	Terry Pratchett	Les Petits Dieux
9	Terry Pratchett	Le Cinquième éléphant
10	Terry Pratchett	La Vérité
11	Terry Pratchett	Le Dernier héros
12	Terry Goodkind	Le Temple des vents

Recherche naturelle

```
1  SELECT *
2  FROM Livre
3  WHERE MATCH(titre)
4  AGAINST ('Petite');
5
6  SELECT *
7  FROM Livre
8  WHERE MATCH(titre)
9  AGAINST ('Petit');
```

id	auteur	titre
3	Daniel Pennac	La Petite marchande de prose

Recherche naturelle

```
1 SELECT *  
2 FROM Livre  
3 WHERE MATCH(auteur)  
4 AGAINST ('Henri');
```

id	auteur	titre
16	Henri-Pierre Roché	Jules et Jim

Recherche naturelle

```
1 SELECT *
2 FROM Livre
3 WHERE MATCH(auteur, titre)
4 AGAINST ('Jules');
5
6 SELECT *
7 FROM Livre
8 WHERE MATCH(titre, auteur)
9 AGAINST ('Jules Verne');
```

id	auteur	titre
14	Jules Verne	De la Terre à la Lune
16	Henri-Pierre Roché	Jules et Jim
15	Jules Verne	Voyage au centre de la Terre

id	auteur	titre
14	Jules Verne	De la Terre à la Lune
15	Jules Verne	Voyage au centre de la Terre
16	Henri-Pierre Roché	Jules et Jim

Recherche avec booléens

```
1  SELECT *  
2  FROM nom_table  
3  WHERE MATCH(colonne)  
4  AGAINST('chaîne recherchée' IN BOOLEAN MODE);
```

Recherche avec booléens

```
1 SELECT *  
2 FROM Livre  
3 WHERE MATCH(titre)  
4 AGAINST ('+bonheur -ogres' IN BOOLEAN MODE);
```

id	auteur	titre
4	Jacqueline Harpman	Le Bonheur est dans le crime

Recherche avec booléens

```
1  SELECT *
2  FROM Livre
3  WHERE MATCH(titre)
4  AGAINST ('"Terre à la Lune"' IN BOOLEAN MODE);
5
6  SELECT *
7  FROM Livre
8  WHERE MATCH(titre)
9  AGAINST ('"Lune à la Terre"' IN BOOLEAN MODE);
10
11 SELECT *
12 FROM Livre
13 WHERE MATCH(titre)
14 AGAINST ('"Terre la Lune"' IN BOOLEAN MODE);
```



id	auteur	titre
14	Jules Verne	De la Terre à la Lune

Recherche avec booléens

```
1 SELECT *  
2 FROM Livre  
3 WHERE MATCH(titre)  
4 AGAINST ('petit*' IN BOOLEAN MODE);
```

id	auteur	titre
3	Daniel Pennac	La Petite marchande de prose
8	Terry Pratchett	Les Petits Dieux

Recherche avec booléens


```
1 SELECT *  
2 FROM Livre  
3 WHERE MATCH(titre)  
4 AGAINST ('+petit* -prose' IN BOOLEAN MODE);
```

id	auteur	titre
8	Terry Pratchett	Les Petits Dieux

Index : en résumé

- Un index est une structure de données qui reprend la liste ordonnée des valeurs auxquelles il se rapporte.
- Un index peut se faire sur une ou plusieurs colonnes ; et dans les cas d'une colonne de type alphanumérique (CHAR, VARCHAR, TEXT, etc.), il peut ne prendre en compte qu'une partie de la colonne (les x premiers caractères).
- Un index permet d'accélérer les recherches faites sur les colonnes constituant celui-ci.
- Un index UNIQUE ne peut contenir qu'une seule fois chaque valeur (ou combinaison de valeurs si l'index est composite, c'est-à-dire sur plusieurs colonnes).
- Un index FULLTEXT (réservé aux tables MyISAM) permet de faire des recherches complexes sur le contenu des colonnes le constituant.

Indexes in Table

Visible	Key	Type	Uniq...	Columns
<input checked="" type="checkbox"/>	 PRIMARY	BTREE	YES	emp_no, ti





Index Details

Drop Index

Key Name:
Index Type:
Allows NULL:
Cardinality:
Comment:
User Comment:

Packed:
Unique:

Columns in table

Column	Type	Nullable	Indexes
 emp_no	int	NO	PRIMARY
 title	varchar(50)	NO	PRIMARY
 from_date	date	NO	PRIMARY
 to_date	date	YES	

EXPLAIN

EXPLAIN

- L'instruction EXPLAIN fournit des informations sur la manière dont MySQL exécute les instructions.
- EXPLAIN fonctionne avec les instructions SELECT, DELETE, INSERT, REPLACE et UPDATE.
- Lorsque EXPLAIN est utilisé avec une instruction explicable, MySQL affiche les informations de l'optimiseur sur le plan d'exécution de l'instruction.

Avec l'aide d'EXPLAIN, vous pouvez voir où vous devez ajouter des index aux tables afin que l'instruction s'exécute plus rapidement en utilisant des index pour rechercher des lignes.

EXPLAIN

```
EXPLAIN SELECT Name FROM country WHERE Code Like 'A%';
```

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	country	NULL	range	PRIMARY	PRIMARY

key_len	ref	rows	filtered	Extra
12	NULL	17	100.00	Using where

<https://dev.mysql.com/doc/refman/8.0/en/explain.html>

<https://dev.mysql.com/doc/refman/8.0/en/explain-output.html>

<https://dev.mysql.com/doc/workbench/en/wb-tutorial-visual-explain-dbt3.html>

