

## Commandes MySQL :

```
SET AUTOCOMMIT = 0
```

```
select @@autocommit ;  
ou  
select @@session.autocommit ;
```

```
SELECT ...FOR UPDATE ...  
LOCK TABLES table ...{READ | WRITE } ...
```

```
UNLOCK TABLES
```

## MySQL: Le choix de moteur

```
mysql> SHOW ENGINES;  
+-----+-----+-----+  
| Engine      | Support | Comment  
+-----+-----+-----+  
| InnoDB      | YES     | Supports transactions, row-level locking, and foreign  
| MRG_MYISAM  | YES     | Collection of identical MyISAM tables  
| BLACKHOLE   | YES     | /dev/null storage engine (anything you write to it dis  
| CSV         | YES     | CSV storage engine  
| MEMORY      | YES     | Hash based, stored in memory, useful for temporary ta  
| FEDERATED   | NO      | Federated MySQL storage engine  
| ARCHIVE     | YES     | Archive storage engine  
| MyISAM      | DEFAULT | Default engine as of MySQL 3.23 with great performanc  
+-----+-----+-----+  
8 rows in set (0.00 sec)
```

```
1 SET storage_engine=NomDuMoteur;  
2
```

```
mysql> SET storage_engine=InnoDB;
```

**MyISAM** Beaucoup lectures / Recherche textuelle  
**InnoDB** Read+Write / Transactions / Accès clé primaire  
**NDB** Petites Transactions, traitements parallèles  
**MEMORY** Uniquement en mémoire

```
1 /* A la création de la table */  
2 CREATE TABLE maTable(  
3 ...  
4 )ENGINE=MonMoteurDeStockage;  
5  
6 /* En modifiant une table déjà créée */  
7 ALTER TABLE maTable ENGINE=UnAutreMoteur;  
8
```

## LE VERROUILLAGE DE TABLE

Les commandes LOCK et UNLOCK permettent de verrouiller et de déverrouiller une ou plusieurs tables en lecture ou en lecture/écriture.

### Syntaxes

```
LOCK TABLES nom_de_table verrouillage [, nom_de_table verrouillage];
```

```
UNLOCK TABLES
```

- **Exemple**

Ouvrez deux sessions clients (mysql et MySQL Query Browser par exemple).

Utilisateur	Autre utilisateur
SELECT * FROM villes; SET AUTOCOMMIT=0; START TRANSACTION; <b>LOCK TABLES</b> villes <b>READ</b> ;  UPDATE villes SET nom_ville = 'Marsiglia' WHERE cp = '13000';  <b>UNLOCK TABLES</b> ; COMMIT;	SELECT * FROM villes; -- <b>OK</b> UPDATE villes SET nom_ville = 'Marsilia' WHERE cp = '13000'; -- <b>KO</b>

Utilisateur	Autre utilisateur
SELECT * FROM villes; SET AUTOCOMMIT=0; START TRANSACTION; <b>LOCK TABLES</b> villes <b>WRITE</b> ;  UPDATE villes SET nom_ville = 'Marsiglia' WHERE cp = '13000';  <b>UNLOCK TABLES</b> ; COMMIT;	SELECT * FROM villes; -- <b>KO</b> UPDATE villes SET nom_ville = 'Marsilia' WHERE cp = '13000'; -- <b>KO</b>

- **Syntaxe**

```
SELECT * FROM nomDeTable WHERE condition FOR UPDATE;
```

- **Exemple**

Ouvrez deux sessions clients (mysql et MySQL Query Browser par exemple).

Utilisateur	Autre utilisateur
SET AUTOCOMMIT=0; START TRANSACTION;  SELECT * FROM pays WHERE id_pays = '033' <b>FOR UPDATE</b> ;  UPDATE pays SET nom_pays = 'FR' WHERE id_pays = '033';  COMMIT;	SET AUTOCOMMIT=0; START TRANSACTION;  UPDATE pays SET nom_pays = 'fr' WHERE id_pays = '033'; -- Attente bloquante (*)  COMMIT;

(\*) UPDATE sur un autre pays c'est OK.