## MYSQL STORAGE ENGINES

**What is a storage engine??**

A storage engine(DataBase Engine) is a software that a Database Management System uses to create, read, update, and delete (CRUD) data from a database.

The default storage engine for MySQL before version 5.5 was MyISAM. For MySQL 5.5 and later, the default storage engine is InnoDB. Choosing the right storage engine is an important strategic decision, which will impact future development.

mysql> SHOW ENGINES;

mysql> show variables like "default\_storage%";

mysql> SET default\_storage\_engine=”myisam”;

Types of MySQL storage engines

1. 1)Transactional storage engines
2. 2) Non-Transactional storage engines

1) Transactional storage engines

* **InnoDB, NDB**

2) Non-Transactional storage engines

* **MyISAM**
* **MEMORY**
* **ARCHIVE**
* **CSV**
* **MERGE**
* **FEDERATED**
* **BLACKHOLE**

Transactional storage engines support transactions like COMMIT, ROLLBACK, SAVEPOINT. The transaction is nothing but a collection of executable SQL statements. And Transactional storage engines supports foreign key constraint.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MyISAM** | | | | **InnoDB** |
| Few DML operations | | | | OLTP with a lot of DML |
| Very fast for selects | | | | Slower for selects |
| Some data can get corrupted | | | | Good crash-recovery facility |
| Cannot be a MASTER server in replication, but can be a SLAVE | | | | Perfect for MASTER server in the replication |
|  | | | | Supports FOREIGN KEY |
|  | | | | Binary logs |
|  | | | | Point-in-time recovery |
| **Feature** | **MyISAM** | **innoDB** |
| Storage limits | 256TB | 64TB |
| Transactions | No | Yes |
| Locking granularity | Table | Row |
| MVCC | No | Yes |
| Geospatial indexing support | Yes | No |
| Full-text search indexes | Yes | No |
| Clustered indexes | No | Yes |
| Data caches | No | Yes |
| Cluster database support | No | No |
| Foreign key support | No | Yes |
| Point-in-time recovery | No | Yes |

## MyISAM

MyISAM is a MySQL default storage engine (Before Mysql 5.5 version). It is a non-transactional storage engine. It doesn't support transactions. It process the insert and select statements as fast as possible. It supports fulltest index supporting. And also table level locking. No foreign key support. It supports 64 indexes per row. Count(\*) works fast than Innodb. We can compress the data here.

mysql>create database exdb;

mysql>use exdb;

mysql>create table emp(id int(10),name varchar(30),sal double(14,2)) engine=”myisam”;

mysql>show tables;

mysql>desc emp;

mysql>show create table emp;

mysql>show columns from emp;

mysql> create table emp2(id int(10),name varchar(30),sal double(14,2),FULLTEXT(name)) engine="MYISAM";

#cd /var/lib/mysql/dbname/

here we can see emp.frm,(It contains structure) emp.MYI (it contains indexex) ,emp.MYD(it contains data)..So take disk space more.

## Innodb

Innodb is a MySQL default storage engine (**From 5.5 version**). It is a transactional storage engine. **It supports transactions and row-level locking. It provides ACID properties**. It processes update queries as fast as possible. Count(\*) little slow than MyISAM. It doesn't support a full-text index. Here we can't compress the data because everything store in a single file(ibdata).

[Atomicity](http://www.service-architecture.com/database/articles/atomicity.html) :a transaction is a unit of operation - either all the transaction's actions are completed or none are.

Consistency: Any work in progress must not be visible to other transactions until the transaction has been committed.

**Isolation:** A transaction should appear to be running by itself, the effects of other ongoing transactions must be invisible to this transaction, and the effects of this transaction must be invisible to other ongoing transaction.

[Durability](http://www.service-architecture.com/database/articles/durability.html): When the transaction is committed, it must be persisted so it is not lost in the event of a power failure. Only committed transactions are recovered during power-up and crash recovery; uncommitted work is rollbacked.

mysql>create table emp(id int(10),name varchar(30),sal double(14,2)) engine=”innodb”;

mysql> create table dept(eid int(10),dname varchar(30),loc varchar(30),foreign key(eid) references emp(1d)) engine=innodb;

#cd /var/lib/mysql/dbname

here we can see the only emp.frm. The data and indexes will be store in the ibdata file. innodb\_file\_per\_table supports from MySQL 5.5 to stores index and data in .ibd file separately for every table.

**alter an engine**

MYSQL> ALTER TABLE EMP2 ENGINE=INNODB;

## MEMORY (formerly known as HEAP)

For this engine, the data will be stored in the memory. When we restart the server we will lose all the data. When we create a table only the .frm file will be store in the data directory. **It doesn't support BLOB, TEXT data types.**

CREATE TABLE test\_mem (name char(30)) ENGINE = MEMORY;

Insert into test\_mem values ('sarath');

Insert into test\_mem values ('venkat');

Insert into test\_mem values ('chandra');

Insert into test\_mem values ('mamata');

Insert into test\_mem values ('anusha');

## CSV

The CSV storage engine stores data in text files using comma-separated values format.

The CSV storage engine is always compiled into the MySQL server and it won’t all null values.

CREATE TABLE test\_csv (id INT NOT NULL, name CHAR(10) NOT NULL) ENGINE = CSV;

INSERT INTO test\_csv VALUES(1,'sarath '),(2,'prakash');

INSERT INTO test\_csv VALUES(3,'sai '),(5,'papu') ,(4,'janvi');

Select \* from test\_csv

Creating a CSV table also creates a corresponding Metafile that stores the state of the table and the number of rows that exist in the table. The name of this file is the same as the name of the table with the extension CSM. **Parition tables and indexing not supported in CSV Engine**.

## ARCHIVE

This engine stores data in zip format. **It does't support primary key**. It is use to stores historical data. The select statements will be very slow because the data will be stored in compressed format.

create table test\_arch(name varchar(30)) engine=ARCHIVE;

insert into test\_arch values ('tom');

insert into test\_arch values ('sushma');

insert into test\_arch values ('ramya');

insert into test\_arch values ('sravya');

insert into test\_arch values ('anusha');

## MERGE

The MERGE storage engine, also known as the **MRG\_MyISAM** engine, is a collection of identical MyISAM tables that can be used as one. “Identical” means that all tables have identical column and index information.

mysql> CREATE TABLE EMP1 (EMPID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, NAME CHAR(20)) ENGINE=MyISAM;

mysql> CREATE TABLE EMP2 (EMPID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,NAME CHAR(20)) ENGINE=MyISAM;

mysql> INSERT INTO EMP1 (NAME) VALUES ('sarath\_One'),('anu\_one'),('mamu\_one');

mysql> INSERT INTO EMP2 (NAME) VALUES ('sairam\_Two'),('prakash\_two'),('Kiran\_two');

mysql> CREATE TABLE total9 (EMPID INT NOT NULL AUTO\_INCREMENT

PRIMARY KEY,NAME CHAR(20) ) ENGINE=MERGE UNION=(EMP1,EMP2);

mysql> INSERT INTO EMP2 (NAME) VALUES ('seshu\_2'),('giri\_2'),('garuda\_2');

mysql> INSERT INTO EMP1 (NAME) VALUES ('sarath\_1'),('venkat­\_1'),('chandra\_1');

## FEDERATED

The FEDERATED storage engine is used to access the Remote MySQL database without using replication or cluster technology.

#ifconfig

**My Server ipaddress 172.16.88.134**

Remote **ipaddress 192.168.65.150**

CREATE TABLE `test9`(`id` int(10),UNIQUE KEY(`name `( varchar 20)))

ENGINE=FEDERATED

CONNECTION='MYSQL:// **servername**:3306/db\_name/test9';

From My Server, I am executing below Command

CREATE TABLE `Table9`(`A` VARCHAR(100),UNIQUE KEY(`A`(30))) ENGINE=FEDERATED

CONNECTION='MYSQL://**192.168.65.150**:3306/TESTDB/test9';

## BLACKHOLE

In this location **/dev/null** storage engine (anything you write to it disappears), This engine you can use in master-slave replication. In some cases, this will give benefits to replication topology.

mysql> create table black1(name char(20)) engine=BLACKHOLE;

Query OK, 0 rows affected (0.01 sec)

mysql> insert into black1 values ('records');

Query OK, 1 row affected (0.00 sec)

mysql> insert into black1 values ('records');

Query OK, 1 row affected (0.01 sec)

mysql> insert into black1 values ('records');

Query OK, 1 row affected (0.00 sec)

mysql> select \* from black1;

Empty set (0.00 sec)