**(1) Show the total amount of storage being used by each database under MySQL… organized by owner.**

mysql> select table\_schema, sum((data\_length+index\_length)/1024/1024) AS MB from information\_schema.tables group by 1;

+--------------------+-----------------+

| table\_schema | MB |

+--------------------+-----------------+

| prod | 298025.72448921 |

| information\_schema | 0.00781248 |

| maatkit | 70.77330779 |

| mysql | 0.66873168 |

| test | 4752.31449127 |

+--------------------+-----------------+

5 rows in set (0.01 sec)

(2) List the indexes for all tables in database “Sales”… and show me any tables which do not have indexes.

(3) If MySQL database service is Not running, where do you look for problems/errors to correct.

(4) If MySQL is not running, how do you find disk storage used by MySQL …storage FREE/avail for MySQL?

(5) List all active connections to running MySQL DB… and what can you know about activity…

(6) Discuss steps to move/copy a MySQL database from ServerA to ServerB ?

(7) Comment on setup of ServerJ and ServerK for load balancing & fail-over operation, so BOTH stay updated?

(8) Describe steps to backup or restore data/tables/index for database “blue” with other databases still running?

**Why does InnoDB store all databases in one file?**

<https://dba.stackexchange.com/questions/15531/why-does-innodb-store-all-databases-in-one-file?noredirect=1&lq=1>

64

The architecture of InnoDB demands the use of four basic types of info pages

Table Data Pages

Table Index Pages

Table MetaData

MVCC Data (to support Transaction Isolation and ACID Compliance)

Rollback Segments

Undo Space

Double Write Buffer (background writing to prevent reliance on OS caching)

Insert Buffer (managing changes to non-unique secondary indexes)

See the Pictorial Representation of ibdata1

By default, innodb\_file\_per\_table is disabled. This causes all four info page types to land a single file called ibdata1. Many people try to spread out the data by making multiple ibdata files. This could lead to fragmentation of data and index pages.

This is why I often recommend cleaning up the InnoDB infrastructure, using the default ibdata1 file and nothing more.

Copying is very dangerous because of the infrastructure under which InnoDB works. There are two basic infrastructures

innodb\_file\_per\_table disabled

innodb\_file\_per\_table enabled

InnoDB (innodb\_file\_per\_table disabled)

With innodb\_file\_per\_table disabled, all these types of InnoDB info live within ibdata1. The only manifestation of any InnoDB table outside of ibdata1 is the .frm file of the InnoDB table. Copying all InnoDB data at once requires copying all of /var/lib/mysql.

Copying an individual InnoDB table is totally impossible. You must MySQL dump to extract a dump of the table as a logical representation of the data and its corresponding index definitions. You would then load that dump to another database on the same server or another server.

InnoDB (innodb\_file\_per\_table enabled)

With innodb\_file\_per\_table enabled, table data and its indexes live in the database folder next to the .frm file. For example, for the table db1.mytable, the manifestation of that InnoDB table outside of ibdata1 would be:

/var/lib/mysql/db1/mytable.frm

/var/lib/mysql/db1/mytable.ibd

System Tablespace ibdata1

All the metadata for db1.mytable still resides in ibdata1 and there is absolutely no way around that. Redo logs and MVCC data also still live with ibdata1.

When it comes to table fragmentation, here is what happens to ibdata1:

innodb\_file\_per\_table enabled: you can shrink db1.mytables with ALTER TABLE db1.mytable ENGINE=InnoDB; or OPTIMIZE TABLE db1.mytable;. This results in /var/lib/mysql/db1/mytable.ibd being physically smaller with no fragmentation.

innodb\_file\_per\_table disabled: you cannot shrink db1.mytables with ALTER TABLE db1.mytable ENGINE=InnoDB; or OPTIMIZE TABLE db1.mytable; because it resides with ibdata1. Running either command actually, make the table contiguous and faster to read and write to. Unfortunately, that occurs at the end of ibdata1. This makes ibdata1 grow rapidly. This is fully addressed in my InnoDB Cleanup Post.

WARNING (or DANGER as the Robot would say in Lost in Space)

If you are thinking of just copying the .frm and .ibd file, you are in line for the world of hurting. Copying the .frm and .ibd file of an InnoDB table is only good if and only if you can guarantee that the tablespace id of the .ibd file matches exactly with the tablespace id entry in the metadata of the ibdata1 file.

I wrote two posts in DBA StackExchange about this tablespace id concept

Table compression in InnoDB? (under the heading 'Restoring Databases')

How to Recover an InnoDB table whose files were moved around

Here is an excellent link on how to reattach any .ibd file to ibdata1 in the event of mismatched tablespace ids : http://www.chriscalender.com/?tag=innodb-error-tablespace-id-in-file. After reading this, you should come to the immediate realization that copying .ibd files is just plain crazy.

For InnoDB, you only need to something this to move

CREATE TABLE db2.mytable LIKE db1.mytable;

INSERT INTO db2.mytable SELECT \* FROM db1.mytable;

to make a copy of an InnoDB table.

If you are migrating it to another DB server, use mysqldump.

With regard to mixing all InnoDB tables from all databases, I can actually see the wisdom in doing so. At my employer's DB/Web hosting company, I have one MySQL Client that has a table in one database whose constraints are mapped to another table in another database within the same MySQL instance. With one common metadata repository, it makes transactional support and MVCC operability possible across multiple databases.

**ERROR 1114 (HY000) the table … is full` with innodb\_file\_per\_table set to autoextend**

<https://dba.stackexchange.com/questions/43503/error-1114-hy000-the-table-is-full-with-innodb-file-per-table-set-to-aut?noredirect=1&lq=1>

**Scheduled optimization of tables in MySQL InnoDB**

<https://dba.stackexchange.com/questions/12201/scheduled-optimization-of-tables-in-mysql-innodb?noredirect=1&lq=1>

**MySQL: sysbench test - InnoDB vs Memory tables**

<https://dba.stackexchange.com/questions/49839/mysql-sysbench-test-innodb-vs-memory-tables?rq=1>

**Mysql Server 5.6 high memory consumption**

**How do you backup a database in MySQl?**

**https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/mysqldump-sql-format.html**