# 数据库备份计划

## 一、计划概览

### 定期进行full backup（完整的数据库备份）。

### 定期进行incremental back（增量备份）。

### 定期对二者的备份文件进行维护。

## 二、计划实施

### 1、实施时间规划：

1.1、对于完整的数据库备份的时间规划，假如数据库每天增加的数据量特别多，则应该每周进行两次。假如数据库数据增量一般，应该每周进行一次足够。

1.2、对于增量备份，最好是每天进行。这样的话可以保证数据库可以恢复到具体的某个时间点。

1.3、二者的备份应该在固定是时间段进行，这样便于管理和组织文件。可以选用晚上业务并不是那么频繁的时刻进行备份。

### 2、实施方式（数据库备份方式）

2.1、完整备份的备份方式：

使用mysqldump进行备份【1】，mysqldump支持数据库不脱机、无阻塞的备份。Mysqldump可以从命令行运行，也可以在mysql workbench管理工具中运行。

使用mysql workbench进行备份，针对数据库引擎是innodb的备份有两种形式：一种是针对每个表生成一个sql文件(separately backup)，一种是数据库的所有信息都放在一个sql文件中(self-contained backup)。前者更容易还原，但是需要的备份时间相对后者更长。后者还原时间更长，备份需求的时间更短。**两种备份方式都可以备份存储过程、事件、触发器。**

2.2、增量备份的备份方式：

**首先，每次启动mysql server都启动 –log-bin 选项，确保数据库执行相关操作过程中会生成二进制文件。【2】**

增量备份可以通过 flush logs或者mysqladmin flush-logs命令执行。

每次运行 flush logs命令，均会产生一个binary log文件。在linux环境中，bin\_log文件应该是位于mysql安装的文件夹的/data文件夹之下。

2.3、备份文件的维护方式：

建议把备份文件存放在其他不易损坏的存储介质中，例如其他的硬盘（这里可以使用raid方式，具体不详述应该采用那种raid方式）。

建议每次备份都把相关的文件上传至远程的服务器，其服务器的位置最好远离数据库主机所在位置，这样可以防止自然灾害。从自然灾害的影响中确保数据安全。

备份文件应该有条理的存放，完整备份应当按照年-月-周的文件夹层次存放。增量备份则应该在符合以上的层次之下，按照进行增量备份时的具体时刻来命名文件。

**每个备份应该要有具体的说明文件，记录备份过程中发生的事件。记录要符合固定的格式要求。**

**2.4、备份方式的具体说明：**

【1】、使用mysqldump进行备份。

mysqldump是mysql官方提供的备份工具，备份方式属于逻辑备份，备份和还原粒度在服务器级别（所有数据库）、数据库级别（特定数据库中的所有表）或表级别可用。无论存储引擎如何，都是如此。备份不包括日志或配置文件，或其他不属于数据库的数据库相关文件。以逻辑格式存储的备份独立于计算机，并且高度可移植。Mysqldump可以实现热备份，即备份过程不需要关闭Mysql服务。所以对现有的app的运行影响不大。

具体的备份方式，详细可见mysql workbench[官方文档](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html)。

【2】、关于mysql启动—log-bin选项的说明：

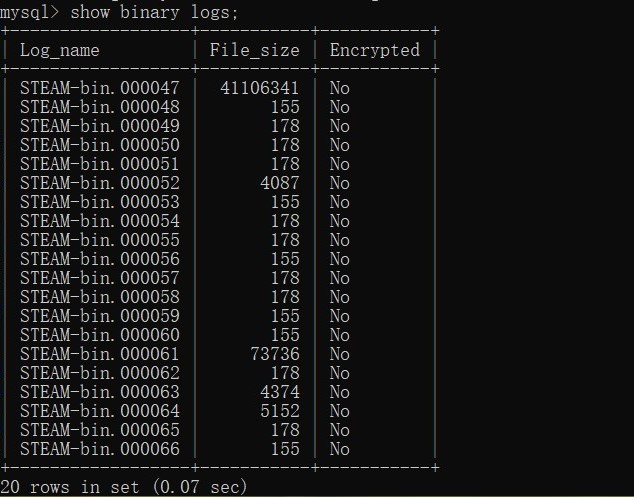
启动mysql的过程中，在命令行中带有 –log-bin参数，即可以使得mysql生成binary log文件，该文件会自动记录每次在数据库执行的操作。通过使用该文件，可以在完整备份复原之后，接着复原自完整备份之后，在其基础之上的数据库的改动。

相关命令演示：

Mysql-> show variables like '%bin%'

检查bin\_log功能是否开启。

Mysql-> show binary logs;



追踪当前的binary log：

假如当前命令出错或者没有输出，则说明过去mysql并没有启动binary log功能。

**使用 mysql-> SET GLOBAL log\_bin = ON;命令，开启bin\_log。**

### 3、数据库还原过程（三个层面的还原）：

3.1、完整备份的还原过程：

使用mysql workbench工具对sql进行导入和执行。**此时separately backup的sql文件可以对单个表进行还原；self-contained sql则可以用来给整个数据库还原。**

3.2、增量备份的还原过程：

使用mysqlbinlog（**UNIX/LINUX命令**），对binary log文件进行解析，对数据库还原。**这样可以达成数据库事务层面的还原。**

3.3、还原过程中的其他操作：

还原过程中，应该仔细阅读数据库备份时记录的日志文件和文件说明。在还原之前，应该对原始数据库进行备份（假如允许的话），确定要还原的时间点之后，应当仔细阅读备份时的备份文件说明文件，排除还原时可能出现的问题。

## 三、问题解答

以下对备份过程中应当考虑的问题进行说明：

### 数据库的恢复层次需求：

1.1、整个数据库服务的恢复，万一服务宕机，需要这种层次的恢复。

1.2、某个数据库的恢复

1.3、某个表的恢复

答：三个层次的恢复，可以参照上文的[恢复过程](#_3、数据库还原过程（三个层面的还原）：)。

### 2、数据库的恢复/备份时间：

* 1. 恢复时间分为两种情况：完整备份的恢复时间和增量备份的恢复时间。前者的时间，与备份的大小有关。原始备份越大，恢复就更加需要时间。增量备份一般意义上是比完整恢复要快的多的。因为只是重新执行备份下的数据库操作。
  2. 数据库的备份时间同理：原始数据越大，消耗的时间也就更多。需要关于备份过程中会对连接数据库的客户端产生什么影响，请参阅【**2.4、备份方式的具体说明】。**

### 3、备份的预处理工作/尾处理工作/备份的加密

3.1、备份之前应该先对已有的环境进行评估，查看过去的日志，看看有没有潜在的问题需要解决。假如现有的环境是有问题的，例如表已经损坏，在做修复之前，应当对修复目标进行备份，然后着手修复，然后再进行备份工作。按照过去设定好的备份文件的存储位置、存储形式进行备份。

3.2、备份之后需要对环境进行清理，删除一些不必要的日志文件。

3.3、备份文件是否需要加密？

假如有安全需求的话那么就需要加密，无论加密形式是本地还是远程加密。**加密之后，会增加总的备份/恢复的时间。**