Discussion-8 different ways to defend against a catastrophic loss

Discussion Topic:

Your corporation has invested heavily in a complex MySQL database. Discuss at least four different ways in which you can defend against a catastrophic loss of this important corporate asset. For each approach, suggest how MySQL utilities can be used to implement your approach.

My Post:

Hello Class,

Protecting against data catastrophic loss is a crucial part of database administration. The following are five strategies that protect against catastrophic data loss.

Regular full database backups (also called logical backups) are often described as the first line of defense against data loss. A full backup implies duplicating and storing the entire structure and data of a database(s), usually on a different hard drive. MySQL provides the *mysqldump* functionality to create full backups. mysqldump is a utility program that generates a SQL script file that can be used to recreate the database (Murach, 2019).
For example:
mysqldump ap > /murach/mysql/ap-2019-01-10.sql -u root -p
The *ap* is the name of the database*, ap-2019-01-10.sql* is the script file, and *-u* (user) the user name (root) -p (password).

Utilize Incremental Backups for Point-in-Time Recovery (PITR). An incremental backup only saves changes made since the last full backup, not the full database. This is often implemented after full database backups have already been performed. With MySQL, after running a full backup, the *mysqlbinlog* program utility can be used to backup the changes made to the database using the MySQL binary log file.
For example:
mysqlbinlog /murach/mysql/bin-log.000001 | mysql -u root -p
bin-log.000001 is the binary log file, and 000001 is a command that tells the program that the file needs to be read only. mysql -u root -p is a MySQL command that allows the utility program to log in to the server.

Creating a replica of a database (called Physical Backups). This method creates a raw copy of the files and directories of the entire database system. A major drawback is that this approach cannot capture data that hasn't been written to disk.

Checking and repairing tables regularly, as database tables can become corrupted, especially after sudden system shutdowns. This can lead to significant data loss if not addressed immediately. In MySQL, the command *CHECK TABLE* can be used to check if a table or (list) tables are damaged. The command works for both InnoDB and MyISAM tables
The REPAIR TABLE command can be used to repair corrupted MyISAM tables, but it does not work for InnoDB tables. For InnoDB table, the program utility *mysqlcheck* can be used to both check and repair tables. For example:
mysqlcheck ap -u root -p
mysqlcheck ap --repair -u root -p

Another important aspect of protecting against catastrophic data loss is to implement secure data export and Import processes. By implementing off-site storage, access control with encryption, regular security testing, and maintaining detailed restoration procedure documentation. This improves data resilience, thus protecting against data loss.

-Alex

References:

Murach, J. (2019). Chapter 19: How to back up and restore a database. *Murach's MySQL (3rd ed.).* Murach Books. ISBN: 9781943872367