You can move databases or tables from the database directory to other locations and replace them with symbolic links to the new locations. You might want to do this, for example, to move a database to a file system with more free space or increase the speed of your system by spreading your tables to different disks.

For InnoDB tables, use the DATA DIRECTORY clause of the CREATE TABLE statement instead of symbolic links, as explained in Section 15.6.1.2, “Creating Tables Externally”. This new feature is a supported, cross-platform technique.

The recommended way to do this is to symlink entire database directories to a different disk. Symlink MyISAM tables only as a last resort.

To determine the location of your data directory, use this statement:

```
SHOW VARIABLES LIKE 'datadir';
```

8.12.2.1 Using Symbolic Links for Databases on Unix

On Unix, symlink a database using this procedure:

1. Create the database using CREATE DATABASE:

   ```
   mysql> CREATE DATABASE mydb1;
   ```

   Using CREATE DATABASE creates the database in the MySQL data directory and permits the server to update the data dictionary with information about the database directory.

2. Stop the server to ensure that no activity occurs in the new database while it is being moved.

3. Move the database directory to some disk where you have free space. For example, use tar or mv. If you use a method that copies rather than moves the database directory, remove the original database directory after copying it.

4. Create a soft link in the data directory to the moved database directory:

   ```
   shell> ln -s /path/to/mydb1 /path/to/datadir
   ```

   The command creates a symlink named mydb1 in the data directory.

5. Restart the server.

8.12.2.2 Using Symbolic Links for MyISAM Tables on Unix

> **Note**
>
> Symbolic link support as described here, along with the --symbolic-links option that controls it, and is deprecated; expect these to be be removed in a future version of MySQL. In addition, the option is disabled by default.

Symlinks are fully supported only for MyISAM tables. For files used by tables for other storage engines, you may get strange problems if you try to use symbolic links. For InnoDB tables, use the alternative technique explained in Section 15.6.1.2, “Creating Tables Externally” instead.

Do not symlink tables on systems that do not have a fully operational realpath() call. (Linux and Solaris support realpath()). To determine whether your system supports symbolic links, check the value of the have_symlink system variable using this statement:

```
SHOW VARIABLES LIKE 'have_symlink';
```