

# Presentation on mysqldump

- Chapter from an upcoming book
- Find online at <http://objectorientedphp.com/mysqldump.pdf>

# mysqldump - a utility program

- mysql - the MySQL client
- mysqladmin - administer MySQL
- mysqlimport - LOAD DATA
- etc

# About the mysqldump Utility

- Copy a development database(s)
- Copy data and/or structure
- Use it with any OS
- Redirect output to file using “>” operator
- Typical use: `mysqldump db_name > file.sql`

# Commonly Used Options (I)

- host, port, user, password
- all-databases
- databases *db\_one db\_two ...*
- no-create-db
- no-create-info
- no-data

# Commonly Used Options (2)

- `opt`
- `where='condition'`
- `xml`

# Dumping a Single Database

- `shell> mysqldump db_name ( > file.sql)`
- `shell> mysqldump db_name tblone tbltwo`
- Only tables from the one named database
- The syntax is flexible
- `shell> mysqldump --other-options db_name tblone`
- `shell> mysqldump db_name tblone --other-options`
- `shell> mysqldump db_name --other-options tblone`

# The --opt Option (I)

- A group of options
- On by default
- In some circumstance you will want to turn them off
- Turn off with the --skip-opt option

# The --opt Option (2)

- add-drop-table
- create-options (MySQL-specific)
- extended-insert (one INSERT statement for all data)
- shell> mysqldump db\_name --skip-opt --add-drop-table
- Order is important

# The --all-databases Option

- Move everything to a different server
- shell> mysqldump --all-database
- Remember that the opt options are on
- Existing tables will be dropped
- The 'mysql' table will be copied
- Maybe use the 'no-create-info' option

# The --databases option

- `shell> mysqldump --databases db_one db_two`
- Again, remember that opt is on by default
- This option turns off the --tables option

# The --tables Option

- Overrides the --databases option
- On by default (not as an option in a config file)
- See slide 6
- This option only makes sense if --databases is set in a configuration file
- Options set a command line have the highest precedence

# The --where Option

- `shell> mysqldump db_name tblone --where='field=value'`
- Quotation marks required when there is white space
- No joins
- Mentioned because there are better ways
- `SELECT` from a view and redirect to file, perhaps

# The --xml Option

- Dump the structure and the data in XML format
- Knowing the options can save you time
- No HTML option
- You probably only want to dump data so use the MySQL client with the --html option

# Use Cases

- Most important use cases already seen
- Dumping data only
- Comparing table data
- Ignoring specific tables
- Standard SQL

# Dumping Data

- Copy data to another RDBMS
- Assuming the same database structure
- You don't want extended INSERT - an opt option
- `shell> mysqldump db_name tblone --skip-opt --no-create-info > output.sql`

# Comparing Table Data

- The same database.table on a different server
- Compare the data in both tables
- `mysqldump db_name tblone --skip-opt --order-by-primary --skip-dump-date --skip-comments > local.sql`
- Do the same for the remote table
- Use a diff utility to compare the files

# Commonly Specified Options

- Specify many options at the command line is cumbersome (as in previous example)
- Add common options to the [mysqldump] section of the config file (my.cnf, my.ini)
- Credentials can be specified in [client]
- If you were to add --skip-opt to the config file override it at the command line with --opt

# Ignoring Specific Tables

- During development there are sometimes interim tables
- Use `--ignore-table=db_name.tbl_name`
- Each table specified separately using dot syntax

# Standard SQL

- Exporting to another RDBMS
- Turn `--create-options`, `--extended-insert` off
- Some are MySQL-specific (i.e. `ENGINE`) as is `extended insert`
- `shell> mysqldump db_name --skip-opt > sqlite.sql`
- May import but no guarantees
- The `--compatible=rdbms_name` may help

# Uploading To a Production Server

- Compatibility issues between different versions
- `/*!40101 SET NAMES utf8 */`
- If you have remote access
- `shell> -h production_server < dump.sql`
- Remote access usually disabled
- Copy the dump file to the host then load it locally

# Using phpMyAdmin

- Dump files shouldn't present any syntactic problems
- Size may be an issue -- limits on uploads
- Copy to the server and then import locally from the command line

# Limits of mysqldump

- Good for server and database backups
- Can be useful for data export and migration
- Exporting a CSV file (usually uses data from more than one table) use `SELECT INTO OUTFILE`
- For data formatted in a `<table>`, use `mysql --html`

# Questions

- [peter@objectorientedphp.com](mailto:peter@objectorientedphp.com)
- Download these slides at <http://objectorientedphp.com/mysqldump.pdf>