

MySQL Client Programs & Table Maintenance

4.1 MySQL Command-Line Tools

mysql Client

Basic Connection:

```
mysql -u root -p -h localhost -P 3306
```

```
mysql --user=root --password --host=localhost --port=3306
```

Executing Queries:

Interactive mode

```
mysql -u root -p
```

Execute single query

```
mysql -u root -p -e "SELECT VERSION();"
```

Execute from file

```
mysql -u root -p < script.sql
```

Batch mode (no formatting)

```
mysql -u root -p -B < queries.sql > output.txt
```

Common Options:

-u, --user=name Username

-p, --password Password (prompt if omitted)

-h, --host=name Server host

-P, --port=number Server port (default 3306)

-D, --database=name Database to use

-e, --execute=statement Execute SQL and exit

-v, --verbose Verbose output

-X, --xml XML output format

-H, --html HTML output format

mysqladmin

Administrative tool for server management:

Check server status

```
mysqladmin -u root -p status
```

Shutdown server

```
mysqladmin -u root -p shutdown
```

Flush tables and logs

```
mysqladmin -u root -p flush-tables
```

```
mysqladmin -u root -p flush-logs
```

Kill process

```
mysqladmin -u root -p kill PROCESS_ID
```

Show processlist

```
mysqladmin -u root -p processlist
```

Create/drop database

```
mysqladmin -u root -p create testdb
```

```
mysqladmin -u root -p drop testdb
```

Show server variables

```
mysqladmin -u root -p variables
```

Show server status

```
mysqladmin -u root -p extended-status
```

mysqldump

Database backup tool:

Full database backup

```
mysqldump -u root -p database_name > backup.sql
```

Specific table

```
mysqldump -u root -p database_name table_name > table_backup.sql
```

All databases

```
mysqldump -u root -p --all-databases > full_backup.sql
```

Multiple databases

```
mysqldump -u root -p --databases db1 db2 db3 > multi_backup.sql
```

Backup options

```
mysqldump -u root -p \
--single-transaction \    # Consistent snapshot without locking
--quick \                 # Avoid loading entire result in memory
--lock-tables=false \     # No table locks (better for InnoDB)
--default-character-set=utf8mb4 \
--triggers \              # Include triggers
--routines \              # Include stored procedures
database_name > backup.sql
```

Compressed backup

```
mysqldump -u root -p database_name | gzip > backup.sql.gz
```

Restoring Backup:

```
mysql -u root -p < backup.sql  
mysql -u root -p database_name < table_backup.sql
```

Restore compressed

```
gunzip < backup.sql.gz | mysql -u root -p
```

mysqlpump

Improved backup tool (MySQL 5.7+):

Parallel backup (faster than mysqldump)

```
mysqlpump -u root -p --parallel=4 database_name > backup.sql
```

Exclude tables

```
mysqlpump -u root -p database_name \  
--exclude-tables=temp_table,log_table > backup.sql
```

Exclude databases

```
mysqlpump -u root -p --exclude-databases=mysql,sys \  
> user_databases.sql
```

Compression

```
mysqlpump -u root -p --compress-output=LZ4 database_name > backup.lz4
```

mysqlslap

Load testing tool:

Simple load test

```
mysqlslap -u root -p --concurrency=10 --iterations=100
```

Test specific database

```
mysqlslap -u root -p \  
  --concurrency=5,10,15 \  # Multiple concurrency levels  
  --iterations=10 \  
  --auto-generate-sql \    # Auto-generate test queries  
  --auto-generate-sql-load-type=read \  
  --number-of-queries=1000 \  
  database_name
```

Custom query load test

```
mysqlslap -u root -p \  
  --concurrency=5 \  
  --iterations=10 \  
  --query="SELECT * FROM sakila.actor WHERE actor_id < 100;" \  
  sakila
```

Compare performance (useful for benchmarking)

```
mysqlslap -u root -p sakila > results_before.txt
```

Modify configuration

```
mysqlslap -u root -p sakila > results_after.txt
```

4.2 Table Maintenance Operations

CHECK TABLE

Checks table for errors:

-- *Basic table check*

```
CHECK TABLE table_name;
```

-- *Multiple tables*

```
CHECK TABLE table1, table2, table3;
```

-- *Extended check (slower, more thorough)*

```
CHECK TABLE table_name EXTENDED;
```

-- *Quick check (faster, less thorough)*

```
CHECK TABLE table_name QUICK;
```

-- *Output:*

-- *Table Op Msg_type Msg_text*

-- *db.tbl check status OK*

ANALYZE TABLE

Updates table statistics used by query optimizer:

-- *Analyze single table*

```
ANALYZE TABLE table_name;
```

-- *Analyze multiple tables*

```
ANALYZE TABLE table1, table2, table3;
```

```
-- Check statistics

SELECT

    TABLE_NAME,
    TABLE_ROWS,
    AVG_ROW_LENGTH,
    DATA_LENGTH,
    INDEX_LENGTH

FROM INFORMATION_SCHEMA.TABLES

WHERE TABLE_SCHEMA = 'database_name';
```

-- Statistics are used by optimizer for query plans

-- Should run periodically, especially after bulk operations

OPTIMIZE TABLE

Reclaims unused space and improves performance:

-- Optimize single table

```
OPTIMIZE TABLE table_name;
```

-- Optimize multiple tables

```
OPTIMIZE TABLE table1, table2, table3;
```

-- For InnoDB, this:

-- 1. Rebuilds table

-- 2. Reclaims fragmented space

-- 3. Defragments table

-- 4. Can take significant time for large tables

-- Check size before and after

```
SELECT  
    TABLE_NAME,  
    ROUND(((DATA_LENGTH + INDEX_LENGTH) / 1024 / 1024), 2) AS size_mb  
FROM INFORMATION_SCHEMA.TABLES  
WHERE TABLE_NAME = 'large_table';
```

REPAIR TABLE

Repairs corrupted tables:

-- *Repair table*

```
REPAIR TABLE table_name;
```

-- *For MyISAM only (rarely used with InnoDB)*

```
REPAIR TABLE table_name QUICK;
```

```
REPAIR TABLE table_name EXTENDED;
```

```
REPAIR TABLE table_name USE_FRM;
```

Table Maintenance Best Practices

```
#!/bin/bash
```

Maintenance script

```
DB_NAME="mydb"  
MYSQL_USER="root"  
MYSQL_PASS="password"
```

Check all tables

```
mysql -u $MYSQL_USER -p$MYSQL_PASS $DB_NAME -e "CHECK TABLE *"
```

```
# Analyze all tables  
mysql -u $MYSQL_USER -p$MYSQL_PASS $DB_NAME -e "ANALYZE TABLE *"
```

```
# Optimize large tables (> 100MB)  
mysql -u $MYSQL_USER -p$MYSQL_PASS $DB_NAME << EOF  
SELECT TABLE_NAME  
FROM INFORMATION_SCHEMA.TABLES  
WHERE TABLE_SCHEMA = '$DB_NAME'  
AND (DATA_LENGTH + INDEX_LENGTH) > 104857600  
EOF
```

4.3 Information Schema Queries

Useful Information Schema Queries

List all tables with size:

```
SELECT  
    TABLE_SCHEMA,  
    TABLE_NAME,  
    ROUND(((DATA_LENGTH + INDEX_LENGTH) / 1024 / 1024), 2) AS size_mb,  
    TABLE_ROWS,  
    ROUND((DATA_LENGTH / 1024), 2) AS data_kb,  
    ROUND((INDEX_LENGTH / 1024), 2) AS index_kb  
FROM INFORMATION_SCHEMA.TABLES  
WHERE TABLE_SCHEMA NOT IN ('mysql', 'performance_schema', 'information_schema')  
ORDER BY (DATA_LENGTH + INDEX_LENGTH) DESC;
```

Find tables by row count:

```
SELECT  
    TABLE_SCHEMA,  
    TABLE_NAME,  
    TABLE_ROWS,  
    ENGINE  
FROM INFORMATION_SCHEMA.TABLES  
WHERE TABLE_SCHEMA = 'database_name'  
ORDER BY TABLE_ROWS DESC;
```

List all columns with data types:

```
SELECT  
    TABLE_SCHEMA,  
    TABLE_NAME,  
    COLUMN_NAME,  
    COLUMN_TYPE,  
    IS_NULLABLE,  
    COLUMN_KEY,  
    COLUMN_DEFAULT  
FROM INFORMATION_SCHEMA.COLUMNS  
WHERE TABLE_SCHEMA = 'database_name'  
ORDER BY TABLE_NAME, ORDINAL_POSITION;
```

Find indexes and their columns:

```
SELECT  
    TABLE_SCHEMA,  
    TABLE_NAME,  
    INDEX_NAME,  
    COLUMN_NAME,  
    SEQ_IN_INDEX,  
    INDEX_TYPE  
FROM INFORMATION_SCHEMA.STATISTICS  
WHERE TABLE_SCHEMA = 'database_name'  
ORDER BY TABLE_NAME, INDEX_NAME, SEQ_IN_INDEX;
```

4.4 Summary: Day 4 Key Takeaways

1. **mysql Client:** Primary tool for interactive queries and scripting
2. **mysqladmin:** Server administration (create/drop DB, kill process, shutdown)
3. **mysqldump:** Backup tool with options for consistency and compression
4. **mysqlpump:** Faster backup with parallel processing
5. **mysqlslap:** Load testing and benchmarking tool
6. **Table Maintenance:** CHECK, ANALYZE, OPTIMIZE, REPAIR operations
7. **Information Schema:** Rich source of metadata queries for diagnostics