

## Chapter 19

# How to back up and restore a database

# Objectives

## Applied

- Back up a database so that it can be restored to any point in time.
- Restore a database to any point in time.
- Export any data that's stored in a database to a file.
- Import data from a file into a database.
- Given one or more InnoDB or MyISAM tables, check the tables to determine if they are corrupted and repair them if possible.

## Knowledge

- Distinguish between a full backup and an incremental backup.
- Distinguish between a tab-delimited file and a comma-delimited file.

## **A strategy for backing up databases**

1. Use the mysqldump program to regularly create full backups of each database.
2. Enable the binary log to create incremental backups.

## **A strategy for restoring databases**

1. Use the mysql program to run the SQL script file for the last full backup.
2. Use the mysqlbinlog program to execute all statements in the binary log that occurred after the last full backup.

## Terms to know

- Full backup
- Incremental backup
- Point-in-time recovery (PITR)

# How to change to MySQL's bin directory

## Using Windows

```
cd /program files/mysql/mysql server 5.6/bin
```

## Using Mac OS X or Unix

```
cd /usr/local/mysql/bin
```

# Common options for the mysqldump program

- `--databases`
- `--all-databases`
- `--single-transaction`
- `--routines`
- `--events`
- `--flush-logs`

# How to run the mysqldump program

## For a single database

```
mysqldump ap > /murach/mysql/ap-2015-02-23.sql -u root -p
```

## For specified databases

```
mysqldump --databases ap ex om mysql >  
/murach/mysql/backup-2015-02-23.sql -u root -p
```

## For all databases

```
mysqldump --all-databases >  
/murach/mysql/all-db-2015-02-23.sql -u root -p
```

# How to run the mysqldump program (continued)

## With additional options

```
mysqldump --databases ap ex om mysql --single-transaction  
--routines --events --flush-logs >  
/murach/mysql/backup-2015-02-23.sql -u root -p
```

## Note for Mac OS X

- You typically need to code a dot and slash (./) before the name of the mysqldump program to specify that it's in the current directory (the bin directory).



## Part of the script for a database backup

```
-- MySQL dump 10.13  Distrib 5.6.18, for Win64 (x86)
--
-- Host: localhost      Database: ap
-- -----
-- Server version      5.6.18-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

## Part of the script for a database backup (cont.)

```
--  
-- Current Database: `ap`  
--  
  
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `ap` /*!40100 DEFAULT  
CHARACTER SET latin1 */;  
  
USE `ap`;  
  
--  
-- Table structure for table `terms`  
--  
  
DROP TABLE IF EXISTS `terms`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `terms` (  
  `terms_id` int(11) NOT NULL,  
  `terms_description` varchar(50) NOT NULL,  
  `terms_due_days` int(11) NOT NULL,  
  PRIMARY KEY (`terms_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

## Part of the script for a database backup (cont.)

```
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `terms`
--

LOCK TABLES `terms` WRITE;
/*!40000 ALTER TABLE `terms` DISABLE KEYS */;
INSERT INTO `terms` VALUES (1,'Net due 10 days',10),(2,'Net due 20
days',20),(3,'Net due 30 days',30),(4,'Net due 60 days',60),(5,'Net
due 90 days',90);
/*!40000 ALTER TABLE `terms` ENABLE KEYS */;
UNLOCK TABLES;
--
-- SQL statement for the table structure and data for all other
-- tables and any triggers associated with those tables
--

--
-- SQL statements for all views, stored procedures, functions,
-- and events
--
```

## Part of the script for a database backup (cont.)

```
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;  
  
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;  
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;  
  
.  
.  
.  
-- Dump completed on 2015-02-23 12:52:15
```

## Advanced options for the mysqldump program

- `--add-drop-table`
- `--add-locks`
- `--disable-keys`
- `--comments`
- `--quote-names`
- `--create-options`
- `--compact`
- `--compress`
- `--delete-master-logs`
- `--force`

## How to use the “skip” prefix to disable an option

`--skip-add-drop-table`

# How to use the mysql program to restore databases

## A single database

```
mysql ap < /murach/mysql/ap-2015-02-23.sql -u root -p
```

## Multiple databases

```
mysql < /murach/mysql/backup-2015-02-23.sql -u root -p
```

## Note for Mac OS X

- You typically need to code a dot and slash (./) before the name of the mysql program to specify that it's in the current directory (the bin directory).

# Common options for the mysqlbinlog program

- `--database=db_name`
- `--start-datetime=datetime`
- `--stop-datetime=datetime`

# How to use mysqlbinlog to execute statements

## For all databases

```
mysqlbinlog /murach/mysql/bin-log.000001 | mysql -u root -p
```

## For a specific database

```
mysqlbinlog --database=ap /murach/mysql/bin-log.000001 |  
mysql -u root -p
```

## For a specific time range

```
mysqlbinlog --start-datetime="2015-02-01 00:00:00"  
/murach/mysql/bin-log.000001 | mysql -u root -p
```

## For all databases using multiple binary log files

```
mysqlbinlog /murach/mysql/bin-log.000001  
/murach/mysql/bin-log.000002 | mysql -u root -p
```



## For all databases using multiple binary log files (Mac OS and Unix only)

```
mysqlbinlog /murach/mysql/bin-log.[0-9]* | mysql -u root -p
```

### Note for Mac OS X

- You typically need to
  1. Begin by coding the sudo command
  2. Code a dot and slash before the name of the mysqlbinlog program and the mysql program
  3. Specify a path to the data directory

## How to convert a binary log file to a SQL file

```
mysqlbinlog /murach/mysql/bin-log.000001 >  
/murach/mysql/bin-log.000001.sql
```

## An excerpt from the converted file

```
BEGIN
/*!*/;
# at 173
#120228 13:02:45 server id 1  end_log_pos 201      Intvar
SET INSERT_ID=126/*!*/;
# at 201
#120228 13:02:45 server id 1  end_log_pos 406      Query
thread_id=144      exec_time=0      error_code=0
use ap/*!*/;
SET TIMESTAMP=1330462965/*!*/;
INSERT INTO vendors VALUES
(DEFAULT, 'Eagle Networks', '1289 Olive Ave.', NULL,
'Fresno', 'CA', '93711', '559-431-7283', 'Shaw', 'Doug',
3, 527)
/*!*/;
# at 406
#120228 13:02:45 server id 1  end_log_pos 433      Xid =
94239
COMMIT/*!*/;
```

## An excerpt from the converted file (continued)

```
# at 433
#120228 13:04:14 server id 1  end_log_pos 510      Query
thread_id=144      exec_time=0      error_code=0
SET TIMESTAMP=1330463054/*!*/;
DROP DATABASE ex
/*!*/;
# at 510
#120228 13:04:58 server id 1  end_log_pos 551      Rotate
to bin-log.000003  pos: 4
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
```

## How to use the mysql program to execute the converted file

```
mysql < /murach/mysql/bin-log.000001.sql -u root -p
```

### Note for Mac OS X

- You typically need to
  1. Begin by coding the sudo command
  2. Code a dot and slash before the name of the mysqlbinlog program
  3. Specify a path to the data directory

# The SELECT statement for exporting data to a file

```
SELECT column_list
INTO OUTFILE file_path
[FIELDS [TERMINATED BY string]
      [ENCLOSED BY char]
      [ESCAPED BY char]
FROM table name
[WHERE search_condition]
[ORDER BY order_by_list]
```

## Note for Mac OS X

- If you get an error that indicates that you don't have permissions to write to the /murach/mysql directory, you can write the file to the /tmp directory and then copy it to the /murach/mysql directory.

# A tab-delimited file

## The statement

```
SELECT *  
INTO OUTFILE '/murach/mysql/vendor_contacts.txt'  
FROM vendor_contacts
```

## The file contents

5	Davison	Michelle
12	Mayteh	Kendall
17	Onandongga	Bruce
44	Antavius	Anthony
76	Bradlee	Danny
94	Suscipe	Reynaldo
101	O'Sullivan	Geraldine
123	Bucket	Charles

# A comma-delimited file

## The statement

```
SELECT *  
INTO OUTFILE '/murach/mysql/vendor_contacts.txt'  
FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\'  
FROM vendor_contacts
```

## The file contents

```
"5", "Davison", "Michelle"  
"12", "Mayteh", "Kendall"  
"17", "Onandong", "Bruce"  
"44", "Antavius", "Anthony"  
"76", "Bradlee", "Danny"  
"94", "Suscipe", "Reynaldo"  
"101", "O'Sullivan", "Geraldine"  
"123", "Bucket", "Charles"
```



# How to use the LOAD DATA statement to import data from a file

## The syntax

```
LOAD DATA INFILE file_path
INTO TABLE table_name
[FIELDS [TERMINATED BY string]
      [ENCLOSED BY char]
      [ESCAPED BY char]]
```

## A tab-delimited file

```
LOAD DATA INFILE '/murach/mysql/vendor_contacts.txt'
INTO TABLE vendor_contacts
```

## A comma-delimited file

```
LOAD DATA INFILE '/murach/mysql/vendor_contacts.txt'
INTO TABLE vendor_contacts
FIELDS TERMINATED BY ','
      ENCLOSED BY '"'
      ESCAPED BY '\\'
```

# The syntax of the CHECK TABLE statement

```
CHECK TABLE table_list option_list
```

## Options for the CHECK TABLE statement

- EXTENDED
- MEDIUM
- QUICK
- FAST
- CHANGED
- FOR UPGRADE

### Note

- The CHECK TABLE statement works only when the server is running.

## A statement that checks a single table

```
CHECK TABLE vendors
```

	Table	Op	Msg_type	Msg_text
▶	ap.vendors	check	status	OK

## A statement that checks multiple tables and views

```
CHECK TABLE vendors, invoices, terms,  
invoices_outstanding
```

	Table	Op	Msg_type	Msg_text
▶	ap.vendors	check	status	OK
	ap.invoices	check	status	OK
	ap.terms	check	status	OK
	ap.invoices_outstanding	check	status	OK

## A statement that uses an option

```
CHECK TABLE vendors, invoices FAST
```

	Table	Op	Msg_type	Msg_text
▶	ap.vendors	check	status	OK
	ap.invoices	check	status	OK

# How to repair a MyISAM table

## The syntax of the REPAIR TABLE statement

```
REPAIR TABLE table_list option_list
```

## Common options for the REPAIR TABLE statement

- QUICK
- EXTENDED

## A statement that repairs a single table

```
REPAIR TABLE vendors
```

## A statement that repairs two tables and uses an option

```
REPAIR TABLE vendors, invoices QUICK
```

## Note

- The REPAIR TABLE statement works only when the server is running.

## How to repair an InnoDB table

1. Use a text editor to add this system variable to the configuration file:

```
innodb_force_recovery=4
```

2. Restart the server, and then use the mysqldump program to back up the database.
3. Remove the innodb\_force\_recovery variable from the configuration file, restart the server, and restore the database to fix the corrupted tables and restore as much data as possible.

# Options for checking tables with mysqlcheck

- `--extended`
- `--medium-check`
- `--quick`
- `--fast`
- `--check-only-changed`
- `--check-upgrade`

# How to use mysqlcheck to check tables

## For a single database

```
mysqlcheck ap -u root -p
```

## For multiple databases

```
mysqlcheck --databases ap ex om -u root -p
```

## For all databases

```
mysqlcheck --all-databases -u root -p
```

## For specified tables within a database

```
mysqlcheck ap vendors invoices -u root -p
```

## For a quick check

```
mysqlcheck ap --quick -u root -p
```

## For an extended check

```
mysqlcheck ap --extended -u root -p
```

## Options for repairing tables with mysqlcheck

- `--repair`
- `--extended`
- `--quick`

## How to use mysqlcheck to repair tables

### For a standard repair

```
mysqlcheck ap --repair -u root -p
```

### For an extended repair

```
mysqlcheck ap --repair --extended -u root -p
```

## Note

- You can only use the mysqlcheck program when the server is running.



# Options for checking a table with myisamchk

- `--extend-check`
- `--medium-check`
- `--check`
- `--fast`
- `--check-only-changed`
- `--force`

# How to use myisamchk to check a table

## For a standard check

```
myisamchk "/ProgramData/MySQL/MySQL Server 5.6/data/ex/engine_sample"
```

## For a medium check

```
myisamchk --medium-check "/ProgramData/MySQL/MySQL Server 5.6/data/ex/engine_sample"
```

## For an extended check

```
myisamchk --extend-check "/ProgramData/MySQL/MySQL Server 5.6/data/ex/engine_sample"
```

## Note

- You should only use the myisamchk program when the server is stopped.

## Options for repairing a table with myisamchk

- `--recover`
- `--quick`
- `--safe-recover`

## How to use myisamchk to repair a table

### For a standard repair

```
myisamchk --recover "/ProgramData/MySQL/  
MySQL Server 5.6/data/ex/engine_sample"
```

### For a quick repair

```
myisamchk --recover -quick "/ProgramData/MySQL/  
MySQL Server 5.6/data/ex/engine_sample"
```

### For an extended repair

```
myisamchk --safe-recover "/ProgramData/MySQL/  
MySQL Server 5.6/data/ex/engine_sample"
```

## A command that checks a table and repairs it if necessary

```
myisamchk -force "/ProgramData/MySQL/  
MySQL Server 5.6/data/ex/engine_sample"
```