The Database Backup Has Finished - What's Next ?

Jörg Brühe

Senior Support Engineer, FromDual GmbH

joerg.bruehe@fromdual.com

FromDual GmbH

www.fromdual.com

Support

Consulting

remote-DBA

Training

About Me

- **Development of a Distributed SQL DBMS:**
  porting to Unix,
  interfacing archiver tools (ADSM, NetWorker)
- **MySQL Build Team:**
  release builds incl. tests, packeting, scripts, ...
- **DBA:**
  MySQL for a web platform
  (master-master-replication)
- **Support Engineer (FromDual):**
  support + remote-DBA for MySQL / MariaDB / Percona
  with or without Galera Cluster; consulting, training

Contents

➡ Database Backup – Which Purpose?

Fitness of the Backup, Verification

Using Production Data Completely

Abiding by Data Privacy Laws

Material for Development

Automation

Why DB Backup

Holds for arbitrary DBMS:

- Data are essential for company / person (financial, legal or emotional aspects)
- Loss of data can totally break operations
- „If data did not get backed up, they cannot be important“

Causes for Data Loss ...

- **Hardware defects**
  drive, controller, power supply, ...
- **Software defects**
  DBMS, operating system, application, ...
- **Mishandling / sabotage**
  `delete from T; commit;` / `rm -fr /`
- **Disaster**
  Fire, flood, accident, ...

Risk Analysis Assumptions

Events considered:

- **individual decision**
- **depend on business goals and strategy**
- **depend on technics used and on location**

Result in requirements for backup
e.g. offsite storage required yes/no?

Further Factors

Individual checks required:

- **Operational procedures?**
  (e.g. orders via paper, phone, or online communication?)
- **Commercial, technical, and legal constraints/requirements?**
  (e.g. replication of data and/or log?)

Result in further requirements on backup

(e.g. point-in-time-recovery required yes/no?
Needs separate log backup yes/no?)

Backup Strategy

Results from technical possibilities, risk assumptions, other factors, and costs:

- **Type, contents, and frequency (physical vs logical; in/excluding log; ...)**
- **Tools used**
- **Placement / storage**
- **Rotation / cleanup period**

Database Backup – Which Purpose?

➡ **Fitness of the Backup, Verification**

Using Production Data Completely

Abiding by Data Privacy Laws

Material for Development

Automation

Purpose of DB Backup

Prevent data loss

- **... by restore (recovery)**
- **... according to requirements**
- **... for all risks (considered)**

Backup has served its purpose ONLY if restore (recovery) is successful

=> Acceptance (QA) criterion of the backup

Schrödinger's Backup

„The status of each backup
is unknown
until it is used
for a restore.“

(Karoly Nagy, talk at „Percona Live Europe 2015“)

Some Risks for Recovery ...

- Backup is not suitable for recovery
- Backup is faulty on creation
  (HW, SW, network, disk full, ...)
- Backup becomes defect or unreachable
  in storage

- Recovery procedure is faulty / non-matching
- Recovery fails due to system state
  (HW, SW, network, disk full, ...)

Verification – How?

- Only a successful recovery testifies that the techniques of backup and recovery are suitable / usable / matching
- Check must be a routine action
- No upper limit to thoroughness: Compare schema, row counts, contents, users, privileges, ... is possible

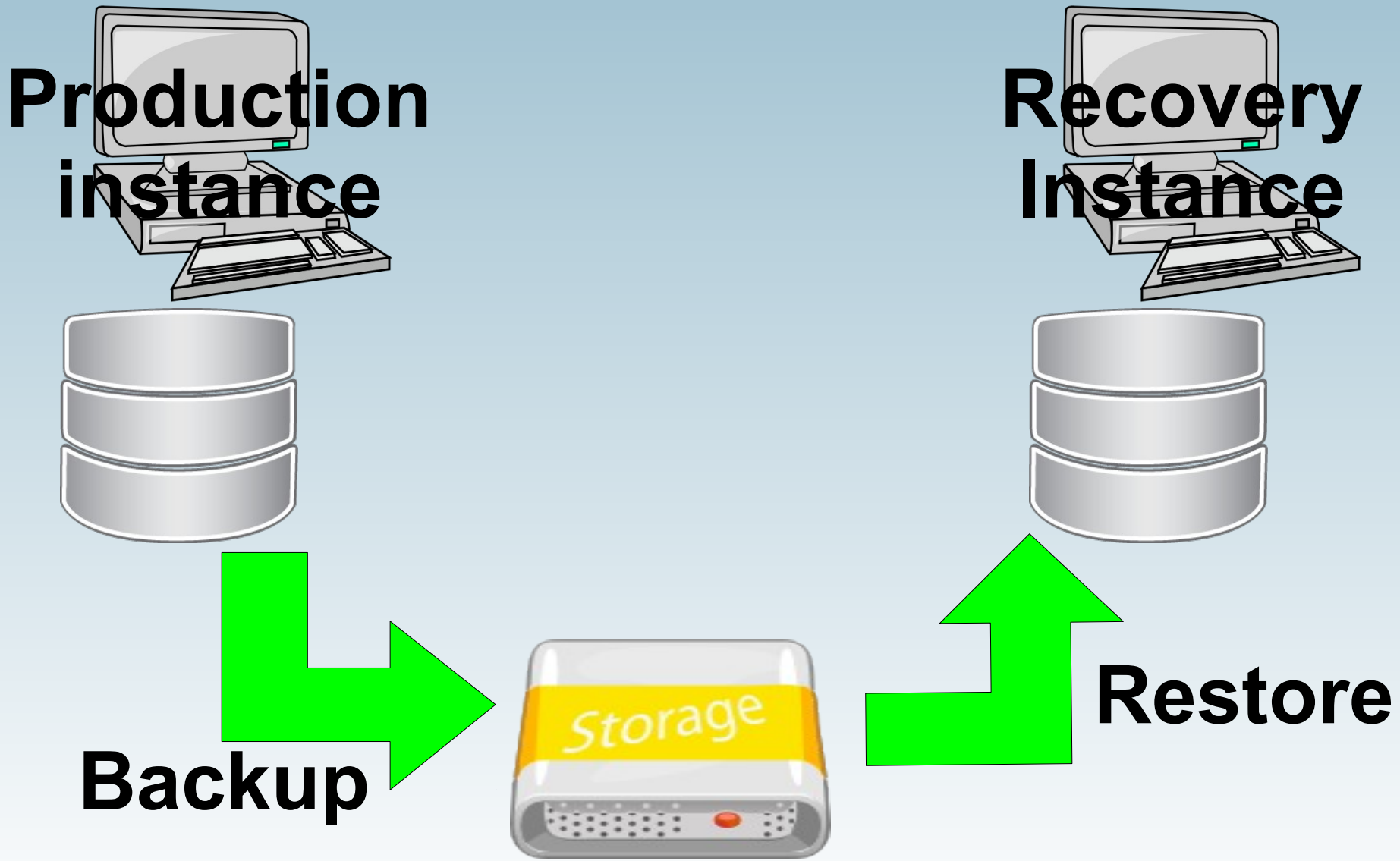
Verification – Where?

- Production backup contains sensitive data
- Production DB must not be affected

Needed:

Separate recovery system in production which is sufficiently large (disk, RAM, ...) for any production DB

Separate Recovery Instance



Database Backup – Which Purpose?

Fitness of the Backup, Verification

➡ **Using Production Data Completely**

Abiding by Data Privacy Laws

Material for Development

Automation

Status After Recovery Test

- Production DB
  - contains Live Data (current)
  - in active use (accesses, load, ongoing changes)
- Recovery DB
  - contains Live Data (as of backup time)
  - on separate system (idle, constant)

Copy of Live Data

- ... subject to data privacy laws
- ... suitable for statistics, billing, ...
- ... allows reliable measurement of e.g. schema changes („create index“, ...)
  (compensate HW differences, if any)

Usage Examples (1)

- Accounting of the last period
  (week, month, ...)
- Statistics / reporting,
  especially if not supported by indexes
- Check for physical errors
  `mysqldump > /dev/null` # complete logical read
- ...

Usage Examples (2)

- Export in different format, e.g.
  - CSV for spreadsheet
  - Mysqldump (plain text) for transfer
- ...

Any read-only non-instantaneous use of live data may be transferred to the recovery server!

Database Backup – Which Purpose?

Fitness of the Backup, Verification

Using Production Data Completely

➡ **Abiding by Data Privacy Laws**

Material for Development

Automation

Data Privacy Catchwords

- Data relating to persons
  e.g. name, address, phone, e-mail, account
- Limited purpose of data collection
- Data sparingness
- Statutory periods for storage / deletion

- Consult with data privacy officer
- If in doubt: get legal advice

Anonymize

Restore instance suports SQL:

- ```
  UPDATE customer
  SET name = 'Doe',
      firstname = 'John'
  WHERE greeting = 'Mr';
  ```

- ```
  UPDATE contact
  SET mail = concat( md5(mail),
                     '@provider.tld');
  ```

Proactive DBA

- **Pessimistic assumption:**
  The others are forgetting data privacy!
- **Remind foreign colleagues**
- **Check the schema yourself,**
  if in doubt: ask
- **Remember your data privacy statement**

Hopefully Needless

- Long term (company) interest
- See various scandals and penalties
- Civil courage, disobedience !
  (aka whistleblower)

Database Backup – Which Purpose?

Fitness of the Backup, Verification

Using Production Data Completely

Abiding by Data Privacy Laws

➡ **Material for Development**

Automation

Developer Desires

- Analyse feature usage
- Test using live data
  Volume, value distribution
- Data volume for tuning
  „Explain“ with/out index
- Test schema change
  „Create Index“, „Alter Table“

Problems / Conflicts

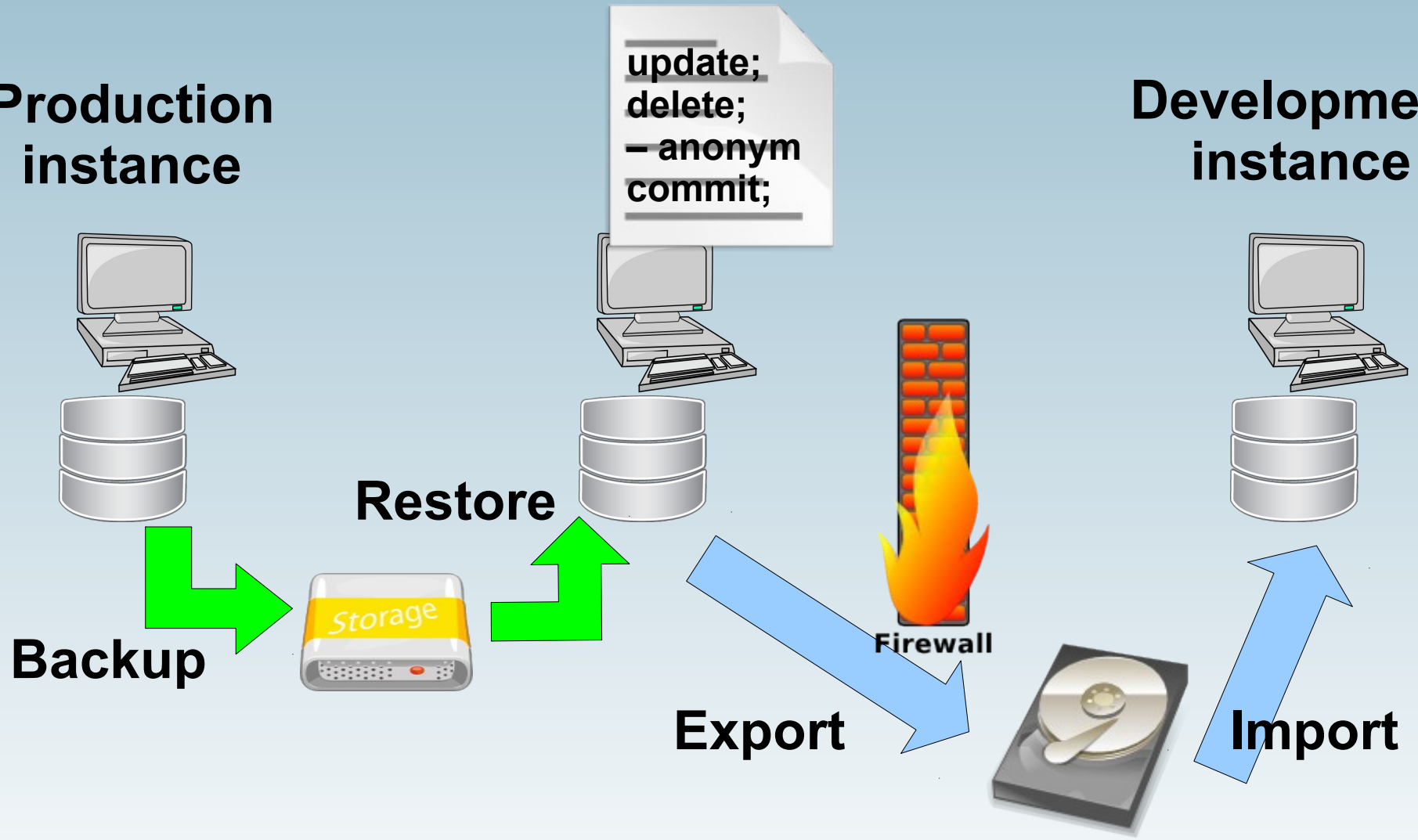
- Data privacy law:
  - Legal oblication
  - Limited purpose
  - Deletion
- Data privacy statement:
  - Company commitment to users
- Data privacy obligation (employee):
  - Confidentiality, ...

Resort / Solution

- In recovery instance
  - delete
  - anonymize
  - selective export
- Deny export
- Accept developer script
  and run on recovery instance
  in controlled DBA environment

Export to Development

www.fromdual.com



Database Backup – Which Purpose?

Fitness of the Backup, Verification

Using Production Data Completely

Abiding by Data Privacy Laws

Material for Development

➡ **Automation**

Preliminaries

- Examples / code for MySQL, please adapt to company rules and translate to different DBMS
- $CONNECT = connection parameters -u USER -pPASSWORD -h HOST
- $DB = selected database (schema)

Automation Principles

- No manual execution of routine tasks
- Success checks contained in script, manual just spot checks
- Error alerts via mail

- No manual configuration / adaption, scripts build lists on their own

Automation: Parts

- Backup (local DBMS instance)
- Purge Log
- Restore (of a given backup)
- Restore wrapper (automated selection)
- DBs/schemas in current restore
- User/password for DB/schema
- Anonymisation script for DB/Schema

Script: Backup (1)

- Backs up DB instance according to strategy
- Generates name(s) with date/timestamp
- Deletes obsolete old backups
- Started by cron

- Separate: monitoring for disk space

Script: Backup (2)

- Helpful: table „backup_history“ with timestamp and exit code
  - „MySQL Enterprise Backup“ does it
  - Replicated to slave?
    MEB 3.9 yes, 3.10 no
    Oracle SR 3-9497158271
- Which instance(s) to backup in replication?

Script: Purge Log (1)

Delete up to last-but-one successful backup:

```
TIMESTAMP=`echo
   "select start_time from
       mysql.backup_history
    where exit_state = 'SUCCESS'
    order by start_time desc limit 1, 1 ;"
| mysql $CONNECT | tail -n1`


echo
   "PURGE MASTER LOGS BEFORE '$TIMESTAMP';"
| mysql $CONNECT
```

Script: Purge Log (2)

Without table – 26 hours back in time:

```
TIMESTAMP=`echo
   "select subtime( now(), '26:00:00')"
| mysql $CONNECT | tail -n1`
```

Script: Restore

- Backup name is passed as a parameter
- Confirmation, if not on restore server
- Deletes all existing DB contents
- Loads (restores, recovers) backup
- Starts DB server process

Script: Restore Wrapper (1)

- Ensures that the backups of all DB instances are subjected to recovery tests
- Runs on restore host only
- Optional parameter: host name
  - If given: determine name of last backup
  - Otherwise: all instances round-robin
- Calls restore script with backup name

Script: Restore Wrapper (2)

Round-Robin on all host names:

```
HOSTLIST=( `ls -d ... |
     grep -iv ...` )
NUMHOSTS=${#HOSTLIST[@]}     # count
DAY=`date '+%j'`          # day of year
INDEX=`expr $DAY % $NUMHOSTS`
BACKUPDIR=${HOSTLIST[$INDEX]}

TIMESTAMP=`ls $BACKUPDIR |
    grep '^20[1-9][0-9]' | tail -1`
```

Code: Which DB/Schema?


```
DBLIST=`cd /DATADIR ; find * -type d`
for DB in $DBLIST
do
    if [ "$DB" = "information_schema"
      -o "$DB" = "mysql"
      -o "$DB" = "performance_schema" ]
    then
        echo "Ignoring database $DB"
        continue
    fi
    ... # next slide
done
```

Code: User/Password for DB

Assumption: User already exists in DB and has the privileges needed for anonymisation

Otherwise: Create after restore

```
DB_S=`echo $DB |
      tr '[:upper:]' '[:lower:]'`
   # 9 chars of DB for User(16)
DB_9=`echo ${DB_S} | cut -c-9`
DB_USR="${DB_9}_anonym"
DB_PWD="${DB_S}#secret#"
```

Code: Handle This DB?

DB without anonymisation user will be skipped

```
mysql -u $DB_USR -p$DB_PWD -e 'show grants;'
RC=$?
if [ $RC -ne 0 ]
then
    echo "'mysql -u $DB_USR -p$DB_PWD'
          fails with code $RC"
    echo "No check / export for this DB."
    continue
fi
```

Code: Search Script


```
for ACTION in check anonym export ; do
   BASE=/path/to/script-${ACTION}-${DB_S}
   SCRIPT=${BASE}.sh
   if [ -x $SCRIPT ] ; then
      bash $SCRIPT -U $DB_USR -P $DB_PWD -N $DB_S
      RC=$?
      echo "$SCRIPT terminated with exit code $RC"
   fi
   SCRIPT=${BASE}.sql
   if [ -r $SCRIPT ] ; then
      mysql -u $DB_USR -p$DB_PWD $DB_S < $SCRIPT
      RC=$?
      echo "$SCRIPT terminated with exit code $RC"
   fi
done
```

Script: Anonymize (1)

Empty all tables unless needed:

```
echo "use information_schema ;
  select TABLE_NAME from TABLES
  where TABLE_SCHEMA = '$DB' and
  TABLE_TYPE = 'BASE TABLE' and
  TABLE_NAME not in ('t1', 't2', 't3') ;" | \
mysql $CONNECT --skip-column-names | \
while read TAB
do
    echo "truncate table $TAB ;"
done | mysql $CONNECT $DB
```

Script: Anonymize (2)

Data with personal information:

```
mysql $CONNECT $DB <<'eof'
UPDATE t1 SET
  payment = 'xxx', birth = '1970-01-01',
  fname = 'John', lname = 'Doe',
  email = 'samplemail@sample.comp',
  ... ;

UPDATE t2 SET
  request = NULL, response = NULL;
eof
```

Code: „select count(*)“


```
# Get a list of all tables, feed it into a loop
# that generates "select count(*)" statements,
# pipe these to another client call for execution

mysql $CONNECT --silent --skip-column-names
 -e "select TABLE_NAME from
       information_schema.TABLES
     where TABLE_SCHEMA = '$DB' and
       TABLE_TYPE = 'BASE TABLE' order by 1" | \
while read TN
do
  echo "select count(*) as '$TN' from $DB.$TN ;"
done | mysql $CONNECT --table $DB
```

Code: „show create table“


```
# Get a list of all tables, feed it into a loop
# that generates "show create table" statements,
# pipe these to another client call for execution

echo 'show tables ;' | \
mysql $CONNECT $DB | tail -n +2 | sort | \
while read TAB
do
    echo "show create table $TAB \G"
done | mysql $CONNECT $DB


# or: mysqldump --no-data
```

Q & A

Questions ?

Discussion ?

We have time for some face-to-face talks...

- **FromDual offers neutral and independent:**
  - **Consulting**
  - **Remote DBA**
  - **Support for MySQL, Percona Server, MariaDB, Galera Cluster**
  - **Training**

www.fromdual.com/presentations