



Hidden Gems in MySQL Shell

Scott Stroz
MySQL Developer Advocate



Obligatory "I Love Me" Slide

- Full-stack developer before we were called "full-stack" developers
- The only constant in my development stack has been MySQL
- Former Paramedic
- I have the best office mate!





What is MySQL Shell?



What is MySQL Shell?

- A modern CLI
- Useful for connecting to and managing MySQL instances
- Multiple modes
 - JavaScript
 - Python
 - SQL
- Many, many useful features





Sandboxes



Sandboxes

- Self-contained MySQL instances running on your local machine
- Are only accessible from **localhost**
- You must have MySQL already installed on the system where you are running MySQL Shell



Photo by [Markus Spiske](#) on [Unsplash](#)

Spinning up a Sandbox

```
dba.deploySandboxInstance(3336)
```



Spinning up a Sandbox

```
A new MySQL sandbox instance will be created on this host in  
/Users/sstroz/mysql-sandboxes/3336
```

```
Warning: Sandbox instances are only suitable for deploying and  
running on your local machine for testing purposes and are not  
accessible from external networks.
```

```
Please enter a MySQL root password for the new instance:
```



Spinning up a Sandbox

```
Deploying new MySQL instance...
```

```
Instance localhost:3336 successfully deployed and started.  
Use shell.connect('root@localhost:3336') to connect to the instance.
```



Spinning up a Sandbox

```
\c root@localhost:3336
```



Spinning up a Sandbox

```
Creating a session to 'root@localhost:3336'  
Please provide the password for 'root@localhost:3336':  
  
Fetching schema names for auto-completion... Press ^C to stop.  
Closing old connection...  
Your MySQL connection id is 12  
Server version: 9.1.0 MySQL Community Server - GPL  
No default schema selected; type \use <schema> to set one.
```



Managing Sandboxes

- Stop Instance - `dba.stopSandboxInstance(3336)`
- Start Instance - `dba.startSandboxInstance(3336)`
- Kill Instance - `dba.killSandboxInstance(3336)`
- Delete Instance - `dba.deleteSandboxInstance(3336)`





Server Upgrade Check



Server Upgrade Check

- Can check your DB schema for issues with upgrading.
- By default compares the MySQL version that matches the Shell version to the instance you are connected.
 - If you are running Shell v. 9.2.0 it will check that the instance you are connected to can upgrade to MySQL v. 9.2.0.



Photo by [Online Marketing](#) on [Unsplash](#)

Running Server Upgrade Check

```
util.checkForServerUpgrade()
```



Running Server Upgrade Check

```
The MySQL server at localhost:33060, version 9.1.0 - MySQL Community Server -  
GPL, will now be checked for compatibility issues for upgrade to MySQL 9.2.0.  
To check for a different target server version, use the targetVersion option.
```



Running Server Upgrade Check

```
1) System variable check for deprecation, removal, changes in defaults values  
or invalid values. (sysVars)  
No issues found
```



Running Server Upgrade Check

2) Issues reported by 'check table x for upgrade' command (checkTableCommand)
Issues reported by 'check table x for upgrade' command

```
mysql_rest_service_metadata.object_fields_with_references - 'utf8mb3' is  
depreciated and will be removed in a future release. Please use utf8mb4  
instead
```



Running Server Upgrade Check

3) MySQL syntax check for routine-like objects (syntax)

The following objects did not pass a syntax check with the latest MySQL grammar. A common reason is that they reference names that conflict with new reserved keywords. You must update these routine definitions and `quote` any such references before upgrading.

These checks were performed using the MySQL 9.2.0 syntax.



Running Server Upgrade Check

```
mysql_innodb_cluster_metadata.v2_set_routing_option - 13:23: syntax error:  
near ''  
mysql_innodb_cluster_metadata.v2_set_routing_option - 13:23: syntax error:  
near ''  
mysql_innodb_cluster_metadata.v2_cs_member_rejoined - 17:64: syntax error:  
near ''  
mysql_innodb_cluster_metadata.v2_cs_member_added - 17:64: syntax error: near  
''
```

More information:

<https://dev.mysql.com/doc/refman/en/keywords.html>



Running Server Upgrade Check

4) Checks for foreign keys not referencing a full unique index (foreignKeyReferences)

Foreign keys to partial indexes may be forbidden as of 8.4.0, this check identifies such cases to warn the user.

`mysql_shorts.hole_score_hole_id_fk` - invalid foreign key defined as
'hole_score(holeId)' references a non unique key at table 'hole'.

`view-demo.FKBA1D7AA0B7E93ABB` - invalid foreign key defined as
'weekcontest(holeId)' references a non unique key at table 'hole'.

`view-demo.match_result_team_hole_hole` - invalid foreign key defined as
'match_result_hole_team(hole_id)' references a non unique key at table
'hole'.



Running Server Upgrade Check

Solutions:

- Convert non unique key to unique key if values do not have any duplicates.
In case of foreign keys involving partial columns of key, create composite unique key containing all the referencing columns if values do not have any duplicates.
- Remove foreign keys referring to non unique key/partial columns of key.
- In case of multi level references which involves more than two tables change foreign key reference.



Running Server Upgrade Check

```
5) Check for deprecated or invalid user authentication methods.  
(authMethodUsage)  
No issues found
```



Running Server Upgrade Check

6) Checks for Spatial Indexes (spatialIndex)

No issues found

Errors: 4

Warnings: 3

Notices: 1

ERROR: 4 errors were found. Please correct these issues before upgrading to avoid compatibility issues.





Advanced Data Dump



Advanced Data Dump



Photo by [Michael Fousert](#) on [Unsplash](#)

- We can dump specific tables, specific schemas, or an entire instance
- Dumps are multi-threaded
 - Default is 4 threads
- Dumped data can be filtered
- Data can be dumped to cloud providers
 - OCI, AWS, Azure, S3 compatible
- Can verify HeatWave MySQL compatibility.



Advanced Dump Commands

```
util.dumpInstance('~/.dumps/example1', {threads: 8})
```



Advanced Dump Commands

```
util.dumpSchemas(['mysql_shorts', '~/dumps/example2', {threads: 8}])
```



Advanced Dump Commands

```
util.dumpTables('mysql_shorts', ['games'], '~/dumps/example3')
```



Advanced Dump Commands

```
util.dumpTables('mysql_shorts', ['games'], '~/dumps/example4',  
  {where: {"mysql_shorts.games": "score ≥ 90"}})
```



Advanced Dump Commands

```
util.dumpInstance("example5", {osBucketName:"database_dumps"})
```

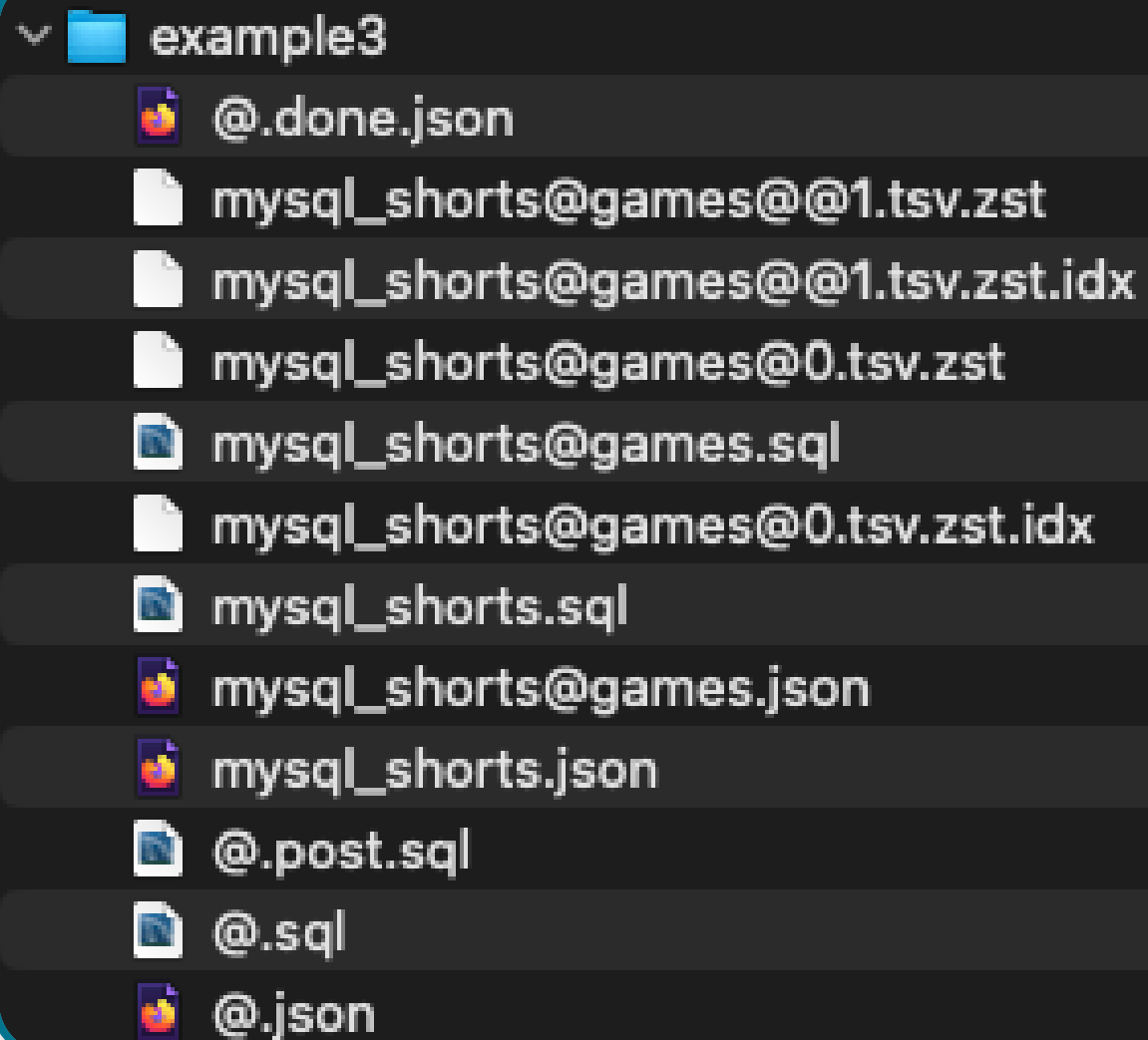


Advanced Dump Commands

```
util.dumpSchemas(['mysql_shorts'], '~/dumps/example6', {ocimds:true})
```



Output



A screenshot of a file explorer window showing the contents of a directory named 'example3'. The directory is expanded, revealing a list of files and subdirectories. The files are listed with their respective icons: a folder icon for 'example3', a file icon for '@.done.json', a file icon for 'mysql_shorts@games@@1.tsv.zst', a file icon for 'mysql_shorts@games@@1.tsv.zst.idx', a file icon for 'mysql_shorts@games@0.tsv.zst', a file icon for 'mysql_shorts@games.sql', a file icon for 'mysql_shorts@games@0.tsv.zst.idx', a file icon for 'mysql_shorts.sql', a file icon for 'mysql_shorts@games.json', a file icon for 'mysql_shorts.json', a file icon for '@.post.sql', a file icon for '@.sql', and a file icon for '@.json'.

- example3
 - @.done.json
 - mysql_shorts@games@@1.tsv.zst
 - mysql_shorts@games@@1.tsv.zst.idx
 - mysql_shorts@games@0.tsv.zst
 - mysql_shorts@games.sql
 - mysql_shorts@games@0.tsv.zst.idx
 - mysql_shorts.sql
 - mysql_shorts@games.json
 - mysql_shorts.json
 - @.post.sql
 - @.sql
 - @.json



Advanced Data Load



Advanced Data Load

- Data dumped using `dumpTables()`, `dumpSchemas()`, and `dumpInstance()` can be loaded using `loadDump()`
- Multi-threaded
 - Default is 4 threads
- Can improve performance by skipping the bin log on load
- Can load data from the cloud
 - OCI, AWS, Azure, S3 compatible
- Create invisible PKs



Photo by [Michael Fousert](#) on [Unsplash](#)

Advanced Load Commands

```
set global local_infile = 'ON';
```



Advanced Load Commands

```
util.loadDump('~/.dumps/example2', {threads: 8})
```



Advanced Load Commands

```
util.loadDump("~/dumps/example2", {skipBinLog:true})
```



Advanced Load Commands

```
util.loadDump("example2", {osBucketName:"database_dumps"})
```



Advanced Load Commands

```
util.loadDump("~/dumps/example2", {createInvisiblePKs: true})
```





Copy Data Between Instances

Copy Data Between Instances

- Data can be moved directly from one instance to another.
- We can copy specific tables, specific schemas, or an entire instance.
- Data can be filtered.



Image by [manuelwagner0](#) from [Pixabay](#)

Copy data Commands

```
set global local_infile = 'ON';
```



Copy data Commands

```
util.copyTables('mysql_shorts', ['games'], 'root@localhost:3336')
```



Copy data Commands

```
util.copySchemas(['mysql_shorts'], 'root@localhost:3336')
```



Copy data Commands

```
util.copyInstance('root@localhost:3336')
```



Copy data Commands

```
util.copyTables('mysql_shorts', ['games'], 'root@localhost:3336',  
  {where: {"mysql_shorts.games": "score ≥ 90"}})
```





Creating a Replica Set

Creating an InnoDB Replica Set

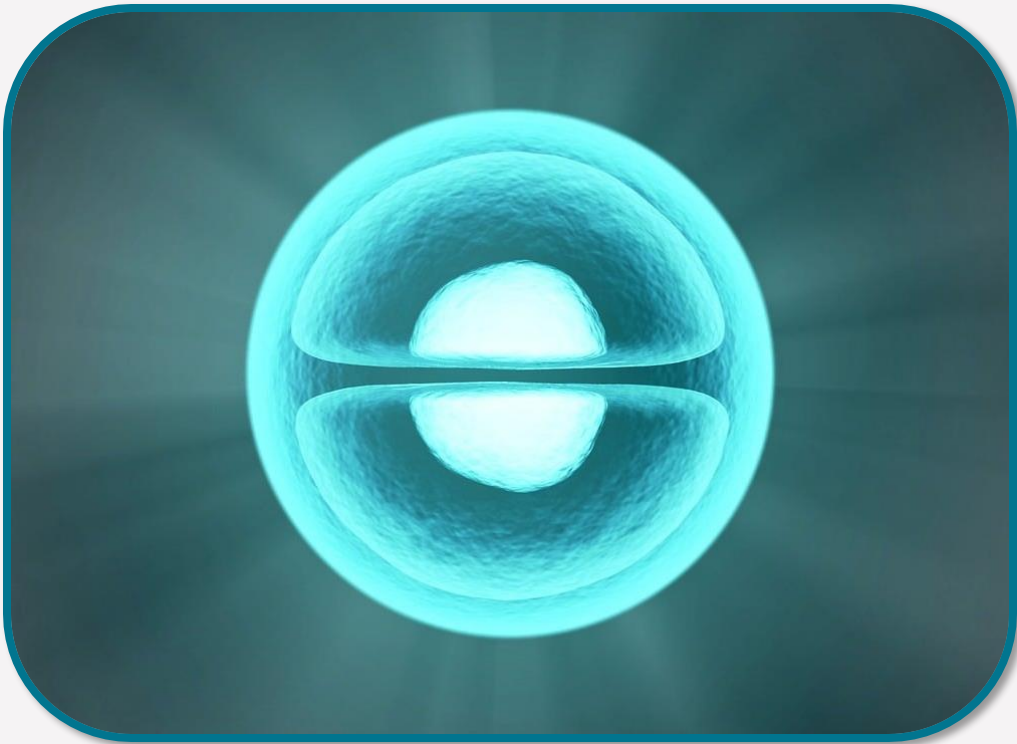


Image by [ar130405](#) from [Pixabay](#)

- Can test setup using sandboxes
- Can be accomplished in two commands



Replica Set Commands

```
var rs = dba.createReplicaSet("demo_set")
```



Replica Set Commands

A new replicaset with instance '127.0.0.1:5555' will be created.

* Checking MySQL instance at 127.0.0.1:5555

This instance reports its own address as 127.0.0.1:5555

127.0.0.1:5555: Instance configuration is suitable.

* Checking connectivity and SSL configuration...

* Updating metadata...

ReplicaSet object successfully created for 127.0.0.1:5555.

Use `rs.addInstance()` to add more asynchronously replicated instances to this replicaset and `rs.status()` to check its status.



Replica Set Commands

```
rs.addInstance('root@localhost:5556')
```



Replica Set Commands

Adding instance to the replicaset...

- * Performing validation checks

This instance reports its own address as 127.0.0.1:5556
127.0.0.1:5556: Instance configuration is suitable.

- * Checking async replication topology...

- * Checking connectivity and SSL configuration...

- * Checking transaction state of the instance...



Replica Set Commands

NOTE: The target instance '127.0.0.1:5556' has not been pre-provisioned (GTID set is empty). The Shell is unable to decide whether replication can completely recover its state.

The safest and most convenient way to provision a new instance is through automatic clone provisioning, which will completely overwrite the state of '127.0.0.1:5556' with a physical snapshot from an existing replicaset member. To use this method by default, set the 'recoveryMethod' option to 'clone'.

WARNING: It should be safe to rely on replication to incrementally recover the state of the new instance if you are sure all updates ever executed in the replicaset were done with GTIDs enabled, there are no purged transactions and the new instance contains the same GTID set as the replicaset or a subset of it. To use this method by default, set the 'recoveryMethod' option to 'incremental'.

Please select a recovery method [C]lone/[I]ncremental recovery/[A]bort (default Clone):



Replica Set Commands

* Updating topology

Monitoring Clone based state recovery of the new member. Press ^C to abort the operation.

Clone based state recovery is now in progress.

NOTE: A server restart is expected to happen as part of the clone process. If the server does not support the RESTART command or does not come back after a while, you may need to manually start it back.

* Waiting for clone to finish...

NOTE: 127.0.0.1:5556 is being cloned from 127.0.0.1:5555

** Stage DROP DATA: Completed

** Clone Transfer

| | | |
|-----------|-------|------|
| FILE COPY | ##### | 100% |
| Completed | | |
| PAGE COPY | ##### | 100% |
| Completed | | |
| REDO COPY | ##### | 100% |
| Completed | | |



Replica Set Commands

```
* Clone process has finished: 76.98 MB transferred in about 1 second (~76.98 MB/s)
```

```
** Changing replication source of 127.0.0.1:5556 to 127.0.0.1:5555
```

```
** Waiting for new instance to synchronize with PRIMARY...
```

```
** Transactions replicated
```

```
##### 100%
```

```
The instance '127.0.0.1:5556' was added to the replicaset and is replicating from  
127.0.0.1:5555.
```

```
* Waiting for instance '127.0.0.1:5556' to synchronize the Metadata updates with the  
PRIMARY...
```

```
** Transactions replicated
```

```
##### 100%
```



Replica Set Commands

```
rs.status()
```



Replica Set Commands

```
{
  "replicaSet": {
    "name": "demo_set",
    "primary": "127.0.0.1:5555",
    "status": "AVAILABLE",
    "statusText": "All instances available.",
    "topology": {
      → "127.0.0.1:5555": {
        "address": "127.0.0.1:5555",
        "instanceRole": "PRIMARY",
        "mode": "R/W",
        "status": "ONLINE"
      },
      → "127.0.0.1:5556": {
        "address": "127.0.0.1:5556",
        "instanceRole": "SECONDARY",
        "mode": "R/O",
        "replication": {
          "applierStatus": "APPLIED_ALL",
          "applierThreadState": "Waiting for an event from Coordinator",
          "applierWorkerThreads": 4,
          "receiverStatus": "ON",
          "receiverThreadState": "Waiting for source to send event",
          "replicationLag": "applier_queue_applied",
          "replicationSsl": "TLS_AES_128_GCM_SHA256 TLSv1.3",
          "replicationSslMode": "REQUIRED"
        },
        "status": "ONLINE"
      }
    },
    "type": "ASYNC"
  }
}
```





Running System Commands



Running System Commands

- System commands can be executed inside MySQL Shell
- Results are displayed in Shell
- Can run commands as **sudo**



Photo by [Michael Fousert](#) on [Unsplash](#)

Running System Commands

```
\system ls -la ~/projects/shell_scripts
```



Running System Commands

```
total 272
drwxr-xr-x 10 sstroz staff 320 May 30 2024 .
drwxr-xr-x 58 sstroz staff 1856 Dec  4 13:24 ..
-rw-r--r--@ 1 sstroz staff 6148 May 30 2024 .DS_Store
-rw-r--r-- 1 sstroz staff 201 May 22 2024 demo1.js
-rw-r--r-- 1 sstroz staff 248 May 22 2024 demo2.js
drwxr-xr-x@ 9 sstroz staff 288 Apr 30 2024 menagerie-db
-rw-r--r--@ 1 sstroz staff 120624 May 24 2024 replica_demo.sql
drwxr-xr-x@ 5 sstroz staff 160 Apr 30 2024 sakila-db
drwxr-xr-x 24 sstroz staff 768 May 30 2024 test_db
drwxr-xr-x@ 3 sstroz staff 96 Apr 30 2024 world-db
```



Running System Commands

```
\! sudo ls /
```



Running System Commands

Password:



Running System Commands

| | | | | |
|------------------|---------|-------|---------|-----|
| .VolumeIcon.icns | Library | bin | home | tmp |
| .file | System | cores | opt | usr |
| .vol | Users | dev | private | var |
| Applications | Volumes | etc | sbin | |





Help System



Help System

- MySQL Shell has a very robust help system
- Every internal object and every internal method has help info.



Photo by [nikko macaspac](#) on [Unsplash](#)

Help System

```
\help
```



Help System

The Shell Help is organized in categories and topics. To get help for a specific category or topic use: `\? <pattern>`

The `<pattern>` argument should be the name of a category or a topic.

The pattern is a filter to identify topics for which help is required, it can use the following wildcards:

- `?` matches any single character.
- `*` matches any character sequence.



Help System

The following are the main help categories:

- AdminAPI The AdminAPI is an API that enables configuring and managing InnoDB Clusters, ReplicaSets, ClusterSets, among other things.
- Shell Commands Provides details about the available built-in shell commands.
- ShellAPI Contains information about the shell and util global objects as well as the mysql module that enables executing SQL on MySQL Servers.
- SQL Syntax Entry point to retrieve syntax help on SQL statements.
- X DevAPI Details the mysqlx module as well as the capabilities of the X DevAPI which enable working with MySQL as a Document Store



Help System

The available topics include:

- The dba global object and the classes available at the AdminAPI.
- The mysqlx module and the classes available at the X DevAPI.
- The mysql module and the global objects and classes available at the ShellAPI.
- The functions and properties of the classes exposed by the APIs.
- The available shell commands.
- Any word that is part of an SQL statement.
- Command Line - invoking built-in shell functions without entering interactive mode.



Help System

SHELL COMMANDS

The shell commands allow executing specific operations including updating the shell configuration.

The following shell commands are available:

- \ Start multi-line input when in SQL mode.
- \connect (\c) Connects the shell to a MySQL server and assigns the global session.
- \disconnect Disconnects the global session.
- \edit (\e) Launch a system editor to edit a command to be executed.
- \exit Exits the MySQL Shell, same as \quit.
- \help (\?,\h) Prints help information about a specific topic.
- \history View and edit command line history.
- \js Switches to JavaScript processing mode.
- \nopager Disables the current pager.
- \nowarnings (\w) Don't show warnings after every statement.
- \option Allows working with the available shell options.
- \pager (\P) Sets the current pager.
- ...



Help System

GLOBAL OBJECTS

The following modules and objects are ready for use when the shell starts:

- audit Audit table management and utilities.
- check Check management and utilities.
- collations Collation utilities
- config MySQL configuration utility.
- dba Used for InnoDB Cluster, ReplicaSet, and ClusterSet administration.
- demo A demo plugin that showcases the shell's plugin feature.
- group_replication MySQL Group Replication management and utilities. A collection of functions to handle MySQL Group Replication without using MySQL InnoDB Cluster (no metadata)
- heatwave_utils Heatwave Utils
- innodb InnoDB management and utilities.
- innodb_cluster MySQL InnoDB Cluster management and utilities.
- legacy_connect Connect to MySQL like old days.
- locks Locks information utilities.
- logs MySQL Logs Utility.
- ...

For additional information on these global objects use: <object>.help()



Help System

EXAMPLES

`\? AdminAPI`

Displays information about the AdminAPI.

`\? \connect`

Displays usage details for the `\connect` command.

`\? checkInstanceConfiguration`

Displays usage details for the `dba.checkInstanceConfiguration` function.

`\? sql syntax`

Displays the main SQL help categories.



Help System

```
\? dumpSchemas
```



Help System

NAME

`dumpSchemas` - Dumps the specified schemas to the files in the output directory.

SYNTAX

```
util.dumpSchemas(schemas, outputUrl[, options])
```

WHERE

`schemas`: List of schemas to be dumped.

`outputUrl`: Target directory to store the dump files.

`options`: Dictionary with the dump options.



Help System

than 8.0.32, all tables at the source server need to have Primary Keys. This needs to be fixed manually before running the dump. Starting with MySQL 8.0.23 invisible columns may be used to add Primary Keys without changing the schema compatibility, for more information see: <https://dev.mysql.com/doc/refman/en/invisible-columns.html>.

In order to use Inbound Replication into an MySQL HeatWave Service DB System instance with High Availability, please see <https://docs.oracle.com/en-us/iaas/mysql-database/doc/creating-replication-channel.html>.

In order to use MySQL HeatWave Service DB Service instance with High Availability, all tables must have a Primary Key. This can be fixed automatically using the `create_invisible_pks` compatibility value.

Please refer to the MySQL HeatWave Service documentation for more information about restrictions and compatibility.



Other Functionality



- Load external scripts
- Create and manage clusters
- Export table data in various formats
- Check MySQL connection
- CLI Integration for scripting
- Retrieve MySQL Instance Diagnostic Information
- Customization
 - Prompt
 - Plugins
 - Startup scripts
- More...





How to reach me...

- **Email:** scott.stroz@oracle.com
- **BlueSky:** [@stroz.dev](https://bsky.app/profile/@stroz.dev)
- **Mastodon:** [@sstroz](https://mastodon.social/@sstroz)
- **LinkedIn:** [scott-stroz](https://www.linkedin.com/in/scott-stroz)





Q&A

Thank You!

