

MySQL for Oracle DBAs

Ahbaid Gaffoor

*This is a living document, the most recent version can be found at
<http://ocpdba.net/mysql/MySQL4OracleDBAs.pdf>*

ahbaid@gmail.com - Version 0.11 Thursday 31st May 2012

Table of Contents

1. Overview:	3
2. Oracle / MySQL Equivalence:	4
3. MySQL Architecture:	5
I. Overview	5
II. Diagram	5
4. MySQL Data Types	6
I. String based types:	6
II. Numeric based Types:	7
III. SQL Mode impact on Data Type Validation	8
Non-Strict Mode Example:.....	8
Strict Mode Example:	9
IV. Date and Time Data Types:	9
4. Storage Engines	10
I. Overview:	10
II. MyISAM:	10
III. InnoDB:	10

1. Overview:

An Oracle DBA is expected to understand all aspects of the database. This includes the files, the background processes, memory, cpu requirements, network connectivity, backups, recovery, high availability, disaster recovery, failover and other areas.

Most Oracle DBAs who have worked only on Oracle database technology try to see other databases through the same lens. This can cause some confusion as not all of the Oracle database architecture fits in the MySQL world and vice versa.

This paper attempts to cover some of the key areas needed for an Oracle DBA to use the MySQL database. It is intended to cover more than the typical install of MySQL and go into some detail on the configuration and maintenance of a MySQL database with a bias towards an Oracle DBA audience.

This document is not intended to be a detailed coverage of MySQL, for details and further explanation, the online documentation is an excellent resource for further reading.

2. Oracle / MySQL Equivalence:

There are some concepts that map between Oracle and MySQL nicely such as table, index and permissions. However, there are some which appear to map but can have very different behavior. This table lists some of the topics that align and some that are unique to each database.

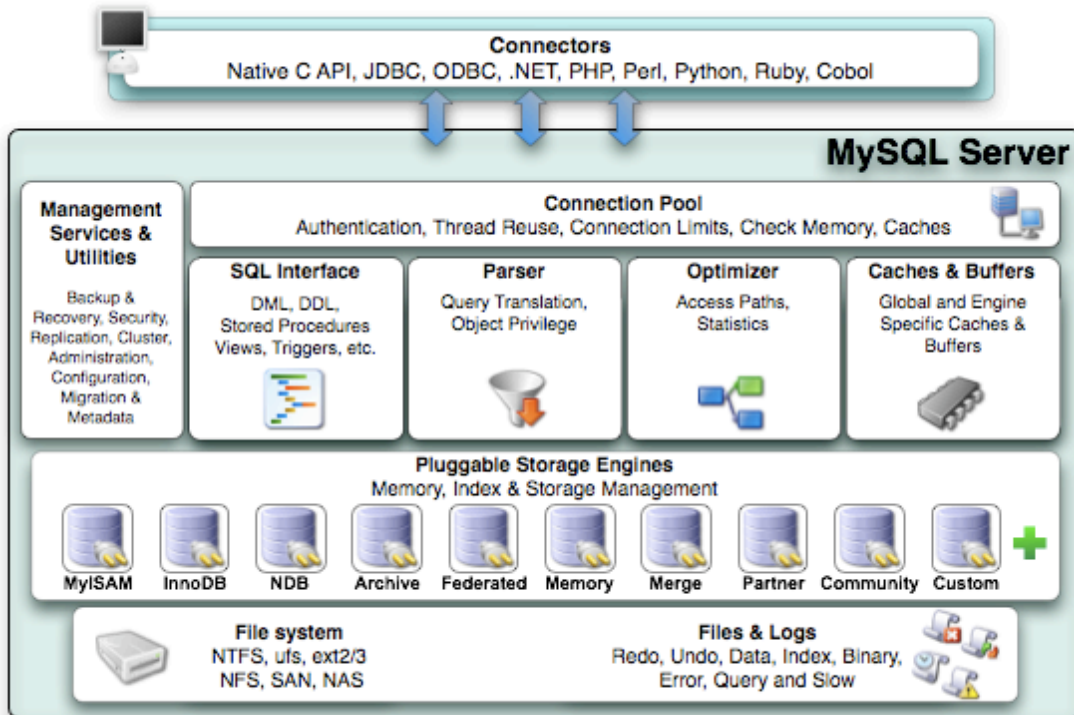
Oracle	MySQL
pfile	/etc/my.cnf
spfile	none
Control file	none
Redo log	Binary Log
Standby	Replication
Data Guard Maximum Protection Mode	Semi-Synchronous Replication
Standby missing redo leads to rebuild	Missed log can be skipped
RAC clustering	NDB Cluster
User	User
Schema	Database
Instance processes	mysql process
User processes	threads
sqlplus	mysql
Listener	port
Alert log	General log, Error log, Slow query log
SHOW PARAMETER	show variables like '%log%';
Log Switch	Flush Logs
-	Flush Privileges
Data Dictionary	performance_schema(5.5.3), information_schema
V\$SESSION, V\$SQL	show processlist;
Explain Plan	Explain Plan for SQL to run (no DBMS_XPLAN like features). SELECT only upto 5.5 DEL, INS, UPD and REPL added 5.6
exp/imp	mysqldump/mysqlimport
Hot Backup	InnoDB(Percona xtrabackup, Oracle Ent. Backup), MyISAM (Flush, read lock)
rlwrap sqlplus	Command history
Custom SQL*Plus prompt: "SET SQLPROMPT"	mysql -prompt
Transaction Serialization	tx_isolation setting
Rollback guaranteed	Rollback can be partial (storage engine)
Trace Dumps	-
Segment/Extent Management	Storage Engine
Shared Server/MTS	Thread Pooling (5.5)
Metalink	bugs.mysql.com
Oracle Open World	Percona MySQL Conference

3. MySQL Architecture:

I. Overview

- a. Connection Manager for Connections
- b. Management Services & Utilities for access, backup and recovery, metadata, administration and configuration.
- c. SQL Interface
- d. Parser
- e. Optimizer
- f. Buffers and Caches (Oracle SGA similarity, not identical)
- g. Storage Engines: Different ways of storing data with very different characteristics and use cases.

II. Diagram



Source: <http://dev.mysql.com/doc/refman/5.5/en/pluggable-storage-overview.html>

4. MySQL Data Types

I. String based types:

- h. CHAR: fixed length up to 255 (2^8-1) characters (8 bit, 1 byte).
- i. VARCHAR: variable length up to 65535 ($2^{16}-1$) characters (16 bit, 2 bytes).
- j. BINARY: up to 255 bytes of binary data.
- k. VARBINARY: up to 65535 bytes of binary data.
- l. [TINY | MEDIUM | LONG]* BLOB: Large amounts of binary data.
- m. [TINY | MEDIUM | LONG]* TEXT: Large amounts of text data.
- n. ENUM: list of up to 65535 ($2^{16}-1$) valid string values, of which one choice is valid.
- o. SET: list of up to 64 valid string values, of which multiple choices are valid.

```
1 drop table if exists customers;
2
3 create table if not exists customers (
4     name varchar(255) not null,
5     address varchar(512) not null,
6     phone varchar(255) not null,
7     email varchar(255),
8     salutation enum('Mr.', 'Mrs.', 'Ms.', 'Dr.', 'Prof.'),
9     contact_method set('email', 'mail', 'phone')
10 );
```

```
mysql> desc customers;
```

Field	Type	Null	Key	Default	Extra
name	varchar(255)	NO		NULL	
address	varchar(512)	NO		NULL	
phone	varchar(255)	NO		NULL	
email	varchar(255)	YES		NULL	
salutation	enum('Mr.', 'Mrs.', 'Ms.', 'Dr.', 'Prof.')	YES		NULL	
contact_method	set('email', 'mail', 'phone')	YES		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> alter table customers change column contact_method contact_by
set('phone', 'mail', 'email') not null default 'mail';
```

```
Query OK, 0 rows affected (0.03 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc customers;
```

Field	Type	Null	Key	Default	Extra
name	varchar(255)	NO		NULL	
address	varchar(512)	NO		NULL	
phone	varchar(255)	NO		NULL	
email	varchar(255)	YES		NULL	
salutation	enum('Mr.', 'Mrs.', 'Ms.', 'Dr.', 'Prof.')	YES		NULL	
contact_by	set('phone', 'mail', 'email')	NO		mail	

```
6 rows in set (0.00 sec)
```

II. Numeric based Types:

- a. Integers:
 - TINYINT: -128 to 127 (1 Byte)
 - SMALLINT: -32,768 to 32,767 (2 Bytes)
 - MEDIUMINT: -8,388,608 to 8,388,607 (3 Bytes)
 - INT: -2,147,483,648 to 2,147,483,647 (4 Bytes)
 - BIGINT: -9,223,372,036,854,775,808 to -9,223,372,036,854,775,807 (8 Bytes)

- b. Floating Point:
 - Approximate values to 7 decimal places.
 - Rounding, possible data loss.
 - FLOAT(M,D) or DOUBLE PRECISION(M,D).
 - M: Total digits including decimals.
 - D: Decimal places.
 - DOUBLE PRECISION has greater accuracy but still rounds.
 - DOUBLE PRECISION uses 8 bytes vs 4 bytes for FLOAT.

- c. DECIMAL: Exact precision, no rounding.

- d. BOOLEAN: Alias for TINYINT storing True(1) or False(0).

- e. Preventing negative numbers using UNSIGNED:
 - HEIGHT DECIMAL(5,1) UNSIGNED
 - Increases positive range, example TINYINT UNSIGNED stores 0 to 256.
 - SQL mode affects behavior!

III. SQL Mode impact on Data Type Validation

SQL Mode affects the results of DML statements. Range violation throws errors in Strict Mode, warnings in non-Strict Mode. The resulting data may not be what is expected.

Non-Strict Mode Example:

```
mysql> create table t1(a decimal(5,1) unsigned, b float(5,1) unsigned);
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> desc t1;
```

Field	Type	Null	Key	Default	Extra
a	decimal(5,1) unsigned	YES		NULL	
b	float(5,1) unsigned	YES		NULL	

```
2 rows in set (0.00 sec)
```

```
mysql> insert into t1(a,b) values(-0.1,-0.2);
Query OK, 1 row affected, 2 warnings (0.00 sec)
```

```
mysql> show warnings;
```

Level	Code	Message
Warning	1264	Out of range value for column 'a' at row 1
Warning	1264	Out of range value for column 'b' at row 1

```
2 rows in set (0.01 sec)
```

```
mysql> show variables like 'sql_mode';
```

Variable_name	Value
sql_mode	

```
1 row in set (0.01 sec)
```

```
mysql> > select @@session.sql_mode;
```

@@session.sql_mode

```
1 row in set (0.00 sec)
```

```
mysql> insert into t1(a,b) values(0.1,0.2);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from t1;
```

a	b
0.0	0.0
0.1	0.2

```
2 rows in set (0.00 sec)
```


Strict Mode Example:

```
mysql> select @@session.sql_mode;
+-----+
| @@session.sql_mode |
+-----+
|                     |
+-----+
1 row in set (0.00 sec)

mysql> set sql_mode='strict_all_tables';
Query OK, 0 rows affected (0.00 sec)

mysql> desc t1;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| a     | decimal(5,1) unsigned | YES  |     | NULL    |      |
| b     | float(5,1) unsigned   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into t1(a,b) values (-0.1,-0.2);
ERROR 1264 (22003): Out of range value for column 'a' at row 1
```

IV. Date and Time Data Types:

- a. YEAR:
 - 4 digit format 1901-2155.
 - 2 digit format 2000-2069 using "00-69" or 1970-1999 using "70-99".
- b. DATE:
 - Year, Month and Day: 1000-01-01 to 9999-12-31.
 - Rounding on insert "0.09" inserted into FLOAT(10,1) rounds to "0.1".
 - No time is stored.
- c. TIME:
 - Stores time only.
 - HHH:MM:SS range from -838:59:59 to 838:59:59.
 - Invalid values are stored as 00:00:00.
- d. DATETIME:
 - DATE and Time: "1000-01-01 00:00" to "9999-12-31 23:59:59".
- e. TIMESTAMP:
 - Records current date and time.
 - Auto update of value for any column changes in that row.

4. Storage Engines

I. Overview:

- Within the overall architecture, MySQL allows the storage layer to be plugged into the database. There are many different storage engines available, each with its own unique characteristics.
- When choosing a storage engine awareness of the features and limitations are critical. Two of the more common storage engines used are MyISAM and InnoDB which will be the current primary areas of focus in this document.
- Other example storage engines include: Memory, CSV, Archive, Blackhole, Merge, Federated, Example and Spider.
- Resource: <http://dev.mysql.com/doc/refman/5.5/en/storage-engines.html>

II. MyISAM:

- The MyISAM storage engine was the default storage engine before version 5.5.5.
- Storage limit of 256TB.
- Maximum 2^{64} rows.
- Maximum 64 indexes per table.
- Maximum 16 columns per index.
- Index file size: $(key_length+4)/0.67$
- Maximum key length 1000bytes. (Re-compile source for higher values)
- Table supported by three files:
 - table.frm – table specification
 - table.MYD – table data
 - table.MYI – table indexes
 - Sits under `<datadir>/<database>/`
- No transactional support.
- Hotbackup reliable only with read only lock.

III. InnoDB:

- The InnoDB storage engine is the default storage engine from version 5.5.5.
- mysql and information_schema databases still use MyISAM and cannot be changed.
- Transactional support.
- Storage limit of 64TB.
- Hot Backups without read only locking