Session 18

An Overview of MySQL 5.7

Objectives

- Explain the process to upgrade MySQL 5.1 to MySQL 5.7
- Describe the new features and enhancements added to MySQL 5.7
- Explain geospatial data and how to use it in MySQL
- Explain various types of MySQL programs
- List and describe spatial data types in MySQL
- Identify JavaScript Object Notation (JSON) data types

Migrating from MySQL 5.1 to MySQL 5.7

❑ Upgrading to MySQL 5.7 allows using new improved features and fixing security vulnerabilities existing in current version.

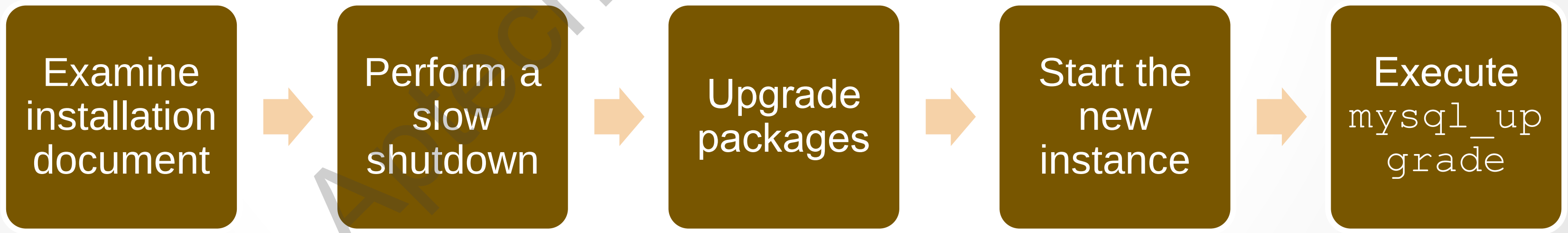
❑ Two ways to upgrade to MySQL 5.7:

- ❖ In-place
- ❖ Logical

In-place Upgrade

- The process includes shutting down the existing instance and replacing its packages with new ones.

- Following figure shows steps of an in-place upgrade:

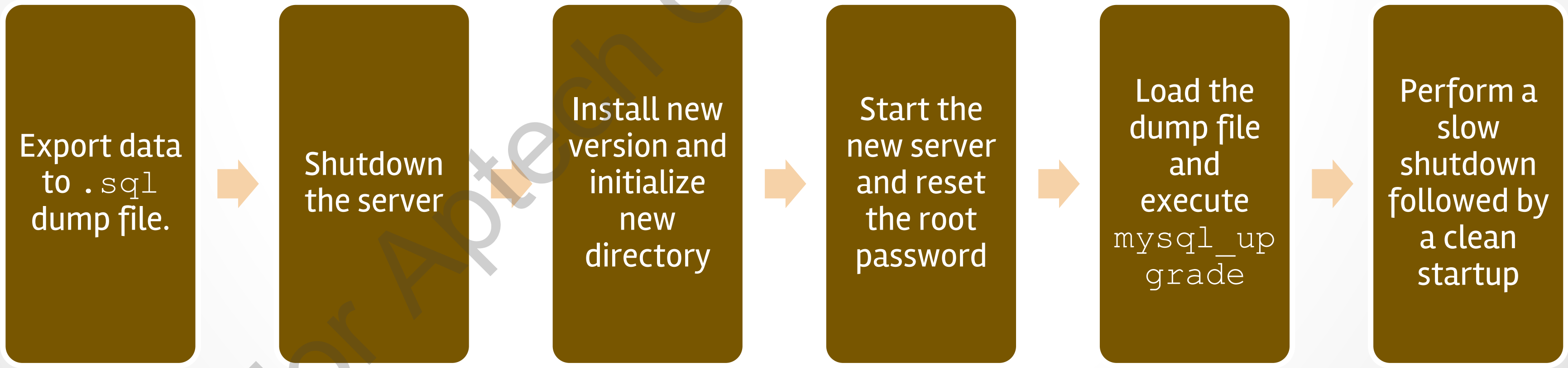


Logical Upgrade

- The process includes exporting existing server data to a dump file, installing the new version, and loading it with the file.

- It is ideal for skipping several versions in between.

- Following figure shows steps of logical upgrade:



Export data to .sql dump file.
Shutdown the server
Install new version and initialize new directory
Start the new server and reset the root password
Load the dump file and execute mysql_upgrade
Perform a slow shutdown followed by a clean startup

Performance Schema

- Is referred to as `PERFORMANCE_SCHEMA` that oversees the server performance at a low level.

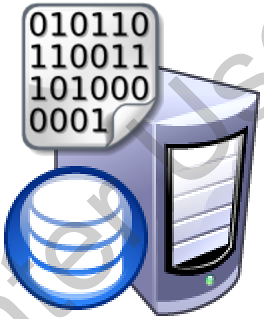
- Contains tables storing the details of status and system variables.

- Focuses mainly on performance data, unlike `INFORMATION_SCHEMA` focusing on metadata.

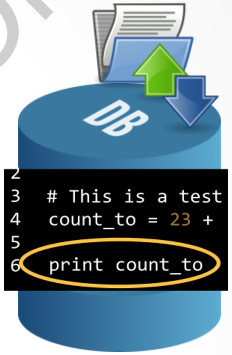
Characteristics of Performance Schema

Server Monitoring

Event Monitoring



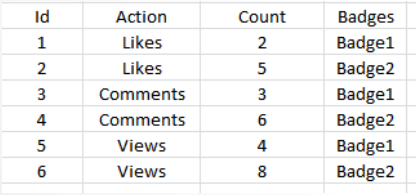
Distinct Events



```
# This is a test
count_to = 23 +

print count_to
```

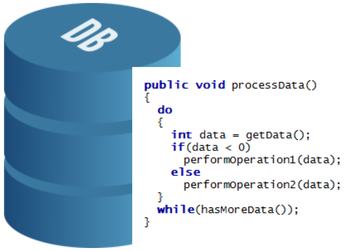
Data Collection via Instrumentation Points



Id	Action	Count	Badges
1	Likes	2	Badge1
2	Likes	5	Badge2
3	Comments	3	Badge1
4	Comments	6	Badge2
5	Views	4	Badge1
6	Views	8	Badge2

Local Temporary Tables

Dynamically Modifiable Configuration



```
public void processData()
{
  do
  {
    int data = getData();
    if(data < 0)
      performOperation1(data);
    else
      performOperation2(data);
  }
  while(hasMoreData());
}
```

Data Collection via Instrumentation

Support on All MySQL Platforms

Benefits of Performance Schema

- ❑ No impact on server’s behavior
- ❑ Non-stop monitoring with least overhead
- ❑ Intact parser
- ❑ Ongoing execution of server code
- ❑ Faster collection of events
- ❑ Easy inclusion of instrumentation points
- ❑ Versioned instrumentation

Tables in Performance Schema

Configuration

- Contain logistical details about monitoring entities.

Object Instance

- Contain instances of instrumented objects.

Events

- Contain the details of monitored events.

History

- Contain details of events occurred.

Summary

- Contain details about completed events.

Other

- Contain rest of the tables.

The sys Schema

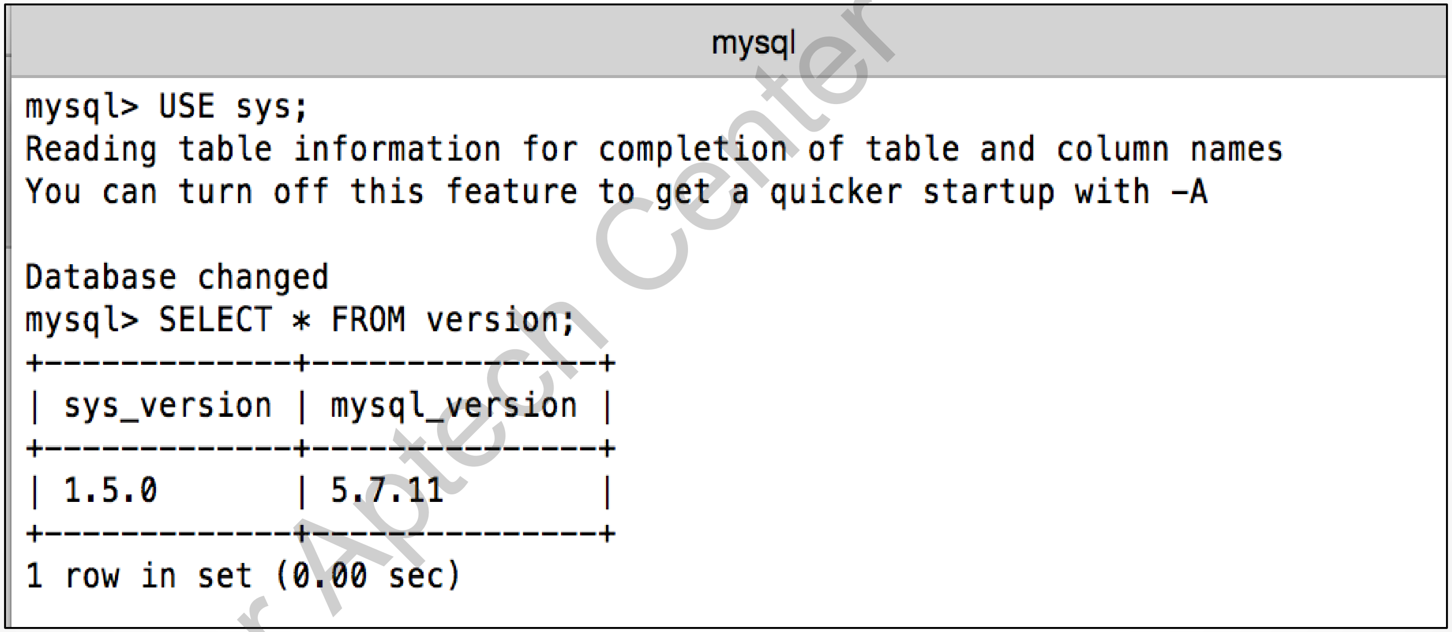
- Is a set of objects, which helps in interpreting data collected by Performance Schema.
- Contains objects such as views, stored procedures, and stored functions.
- Is installed by default while initializing the data directory.
- Has objects with the DEFINER as `'mysql.sys'@'localhost'`.

Prerequisites for Using the sys Schema

- Enabled Performance Schema
- SELECT, EXECUTE, INSERT, UPDATE, PROCESS, and other privileges on different objects
- Enabled Consumer and Instruments of Performance Schema

Using the sys Schema 1-2

- Following figure shows how to set the `sys` schema as default one:



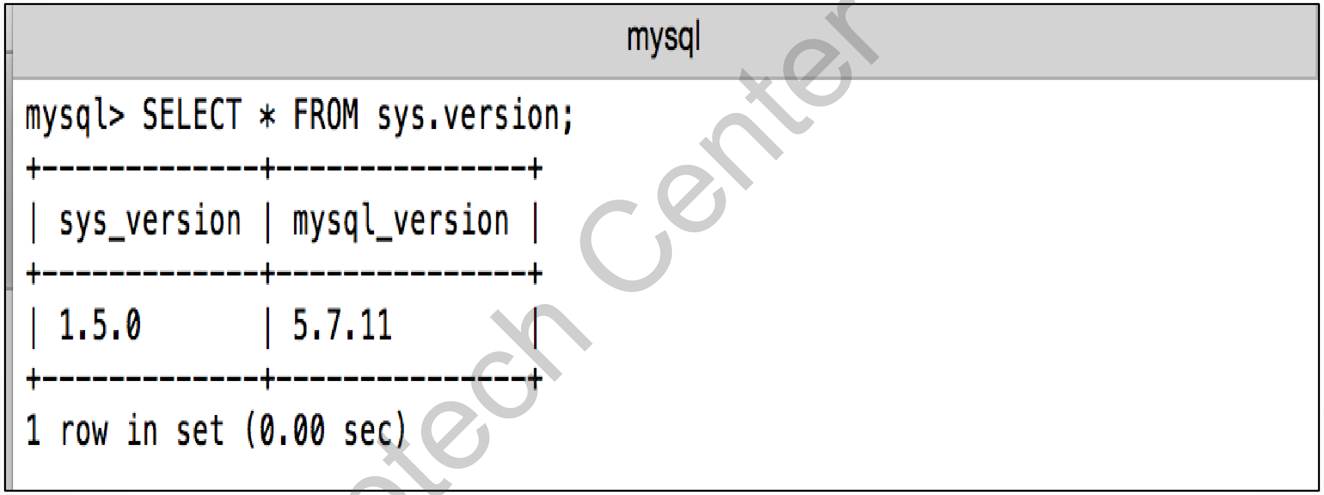
mysql

```
mysql> USE sys;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM version;
+-------------+---------------+
| sys_version | mysql_version |
+-------------+---------------+
| 1.5.0       | 5.7.11        |
+-------------+---------------+
1 row in set (0.00 sec)
```

Using the sys Schema 2-2

❑ Following figure shows how to use object reference through the schema name:



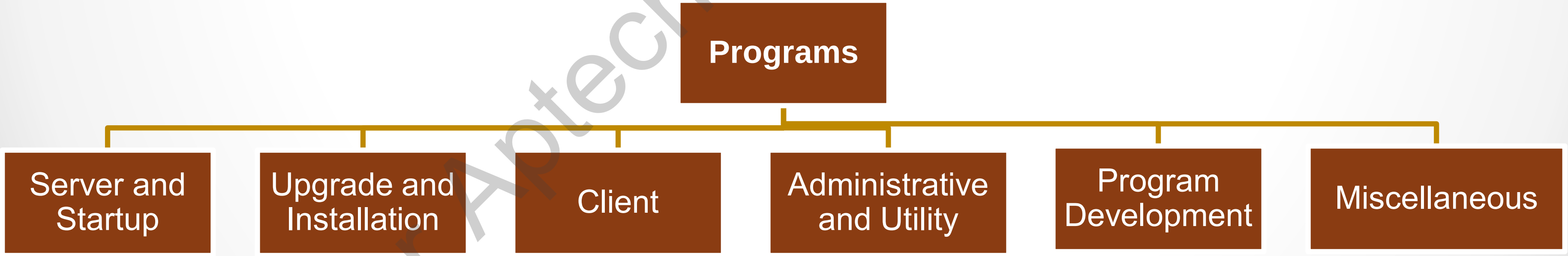
```
mysql> SELECT * FROM sys.version;
+-------------+---------------+
| sys_version | mysql_version |
+-------------+---------------+
| 1.5.0       | 5.7.11        |
+-------------+---------------+
1 row in set (0.00 sec)
```

Security Enhancements in MySQL 5.7

- No compulsion for the `mysql.user` table to hold a plugin column value
- Conversion of accounts using the `mysql_old_password` plugin for using `mysql_native_password` for authentication
- Setup of automatic password expiry policy
- Secure connections through Secure Sockets Layer (SSL) and Rivest Shamir Adleman (RSA) certificate
- Secure upgrades by default through `mysqld -initialize`

MySQL Programs

- ❑ Are mostly used on the command prompt.
- ❑ Have different options, which are seen by using `--help` option.
- ❑ Following figure shows the types of programs available:



Server and Startup Programs 1-2

Server and its Startup Program	Description
`mysqld`	Is the key program required for most MySQL installation tasks.
`mysqld_safe`	Is a server startup script, which starts mysqld.
`mysqld_server`	Is a startup script used on systems for starting and stopping the server.
`mysqld_multi`	Is a startup script useful for starting or halting multiple servers.

Server and Startup Programs 2-2

Installation- or Upgrade-related Program	Description
`comp_err`	Compiles `errmsg.sys` that `mysqld` uses to identify the error message.
`mysql_install_db`	Initializes the data directory, produces `mysql` database, configures `InnoDB`, and installs sys schema.
`mysql_plugin`	Configures server plugins and allows enabling or disabling plugins.
`mysql_secure_installation`	Boosts the security of installation.
`mysql_ssl_rsa_setup`	Produces security certificate and key pairs if they are missing.
`mysql_tzinfo_to_sql`	Loads the tables of time zone in the `mysql` database.
`mysql_upgrade`	Examines tables for incompatibility and fix them, and updates the grant tables.

Client Programs

Installation- or Upgrade-related Program	Description
`mysql`	Returns American Standard Code for Information Interchange (ASCII) table or output separated by tab.
`mysqladmin`	Performs administrative tasks.
`mysqlcheck`	Analyzes, fixes, and optimizes tables.
`mysqldump`	Dumps single or multiple databases into a text, SQL, or XML file for backup.
`mysqlimport`	Imports text files into their corresponding tables.
`mysqlpump`	Dumps a single or multiple databases into an SQL file.
`mysqlshow`	Displays details about databases, indexes, columns, and tables.
`mysqlslap`	Imitates client load and state timings for each stage.

Administrative and Utility Programs

Administrative and Utility Program	Description
innochecksum	Is an offline tool returning checksums for InnoDB files.
myisam_ftdump	Shows details about full-text indexes in MyISAM tables.
myisamchk	Tests, fixes, and optimizes MyISAM tables containing .MYD and .MYI files.
myisamlog	Displays data existing in a MyISAM log file.
myisampack	Compresses MyISAM tables.
mysql_config_editor	Saves authentication credentials in the encrypted file, .mylogin.cnf.
mysqlbinlog	Reads from a binary log for recovering from a crash.
mysqldumpslow	Reads and sums up the matter existing in a slow query log.

Program Development Utilities

Program Development Utility	Description
`mysql_config`	Generates option values required while compiling client programs.
`my_print_defaults`	Displays options and the option files' groups.
`resolve_stack_dump`	Resolve a numeric dump file to symbols.

Miscellaneous Programs

Miscellaneous Program	Description
lz4_decompress	Decompresses the mysqlpump output, which was generated using LZ4 compression.
perror	Infers the MySQL error codes or system.
replace	Replaces string in the input text.
resolveip	Transforms an IP address to its corresponding host name and vice-versa.
zlib_decompress	Decompresses the mysqlpump output, which was generated using ZLIB compression.

Spatial Data

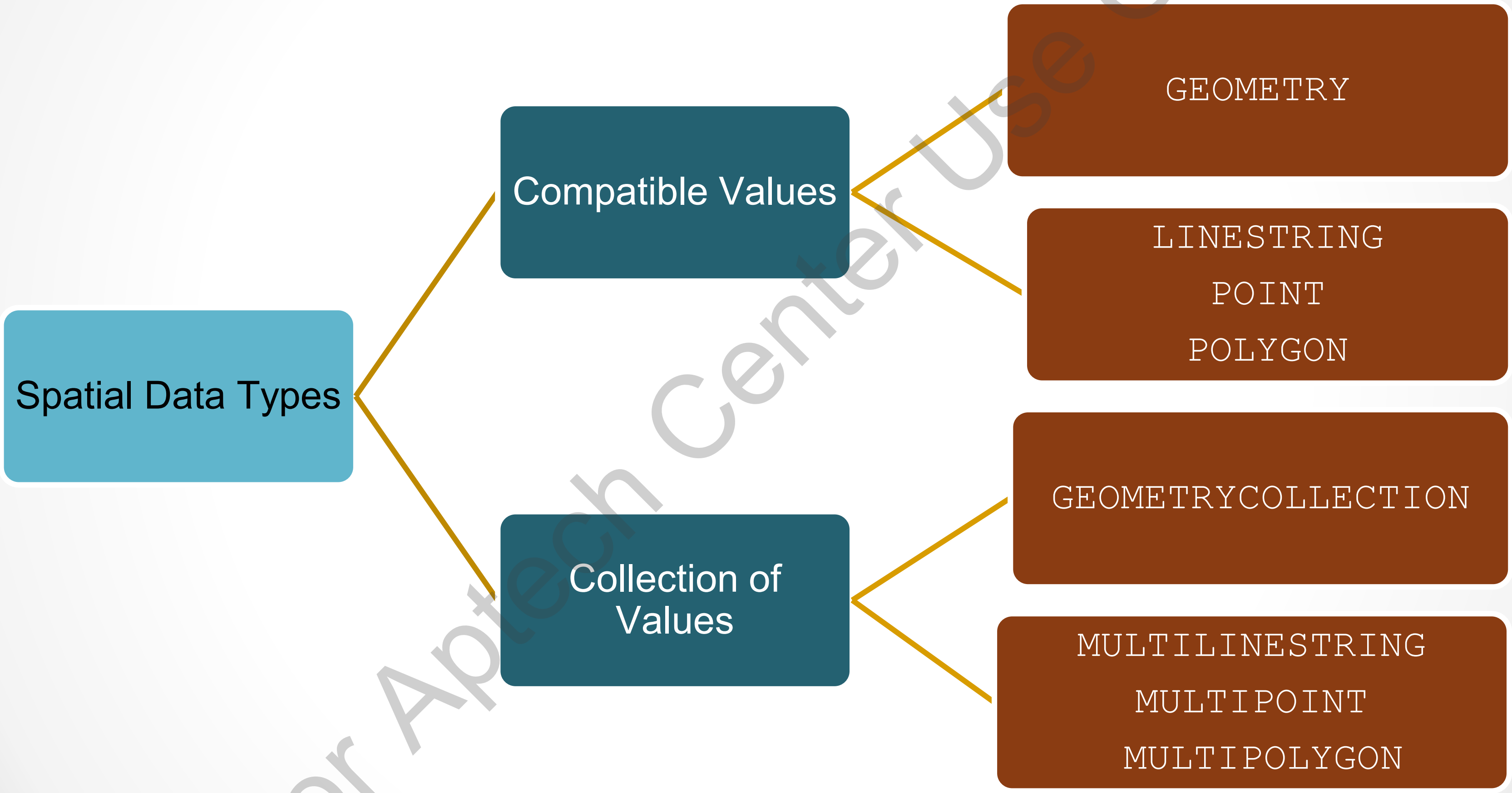
- ❑ Represents geometrical or geospatial data, which refers to a point or a collection of points indicating a location.

- ❑ Is handled as per OpenGIS standard specification released by Open Geospatial Consortium (OGC).

- ❑ Is employed as a Geometry data type and is useful for mapping locations.

- ❑ Is managed through spatial data types and functions, which are only for `NDB`, `InnoDB`, `MyISAM`, and `ARCHIVE` tables.

Spatial Data Types



- Spatial Data Types
  - Compatible Values
    - GEOMETRY
    - LINESTRING
POINT
POLYGON
  - Collection of Values
    - GEOMETRYCOLLECTION
    - MULTILINESTRING
MULTIPOINT
MULTIPOLYGON

OpenGIS Model

- A geometric object is linked to the coordinate space and belongs to a geometry class.

- Geometry is the base class.

- Hierarchy of classes:
  - `Geometry (noninstantiable)`
  - `Point`
  - `Curve`
    - `LineString`
  - `Surface`
    - `Polygon`
  - `GeometryCollection`
    - `MultiPoint`
    - `MultiCurve`
    - `MultiSurface`

Spatial Data Formats

- Two standard data formats for representing geometry objects in MySQL:
  - **Well-Known Text (WKT)** for exchanging geometry values in ASCII form
  - **Well-Known Binary (WKB)** for exchanging values as binary streams denoted by BLOB type

- MySQL, however, does not store geometry values in these formats. Thus, there are functions for converting data into the desired format.

WKT Data Format 1-2

Geometry Object	WKT Representation	Description
`Point`	`POINT(5 10)`	Is represented with X and Y coordinates separated with space and not comma like in the `SQL Point()` function.
	`mysql> SELECT ST_X(Point(5, 10));`	Generates the `Point` object through the `Point()` function and fetches its X-coordinate.
	`mysql> SELECT ST_X(ST_GeomFromText('POINT(5 10)'));`	Uses the WKT format, which is converted to a `Point` object by using the spatial function.

WKT Data Format 2-2

Geometry Object	WKT Representation	Description
`Linestring`	`LINESTRING(0 0, 10 10, 20 25, 50 60)`	Is represented using point coordinate pairs, each followed by a comma.
`Polygon`	`POLYGON((0 0,20 0,20 20,0 20,0 0),(7 7,9 7,7 9,7 9,7 7))`	Is represented using an exterior ring as well as an interior ring.
`MultiPoint`	`MULTIPOINT(0 0, 30 30, 70 70)`	Is represented using point coordinate pairs.

	`ST_MPointFromText('MULTIPOINT (4 4, 5 5, 6 6)')`
	`ST_MPointFromText('MULTIPOINT ((4 4), (5 5), (6 6))')`
		Accepts WKT format of `MultiPoint` values, wherein individual point pair is in parentheses and separated by a comma.

WKB Data Format

- ❑ Accepts 1-byte and 4-byte unsigned integers along with 8-byte double-precision numbers.

- ❑ For instance, POINT(1 0) is represented as a series of 21 bytes, which is split into the following components:
  - ❖ **Byte Order**: 01
  - ❖ **WKB Type**: 01000000
  - ❖ **X –coordinate with Hexadecimal Digit and Double-precision Value**: 000000000000F03F
  - ❖ **Y-coordinate with Hexadecimal Digit and Double-precision Value**: 0000000000000000

Using Spatial Data

- Following code snippet creates the `mobile` table with a spatial column named `geodata`:

```
CREATE TABLE mobile (geodata GEOMETRY);
```

- Following code snippet inserts WKT values:

```
INSERT INTO mobile VALUES
(ST_GeomFromText('POINT(10 10)'));
```

- Following code snippet retrieves data in the WKT and WKB formats respectively:

```
SELECT ST_AsText(geo) FROM mobile1;
SELECT ST_AsBinary(geo) FROM mobile1;
```

JSON Data Type

- ❑ Allows accessing data from a stored JSON document, which is converted into an internal binary format for fast reading of elements.
- ❑ Facilitates automatic validation of JSON files added to JSON columns and optimized storage format.

Example of JSON data:

```
{
 "code": "DE",
 "name": "Germany",
 "population": "81802257"
}
```

{JSON}

Using JSON Data Type

- ❑ Following code snippet inserts a value in JSON column:

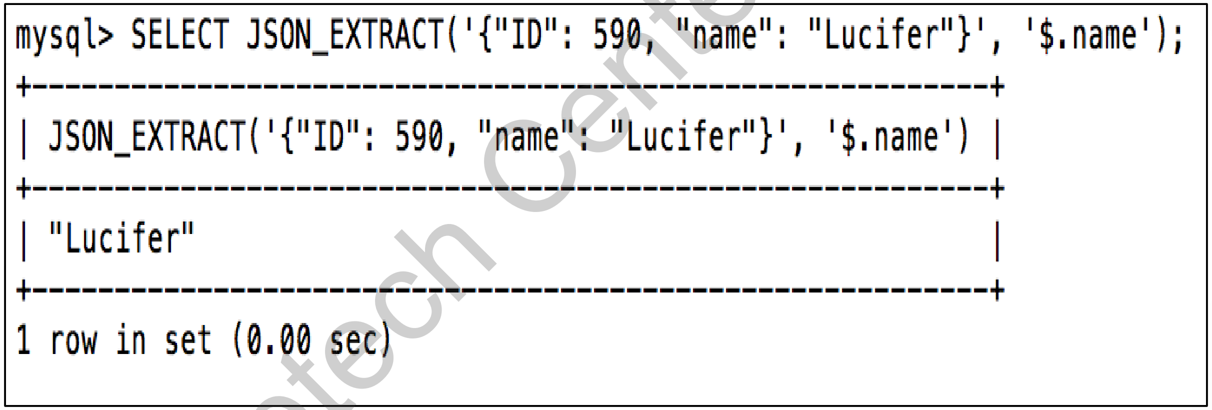
```
CREATE TABLE Book (bdoc JSON);
INSERT INTO Book VALUES('{"Edition": "V1.1",
"Author": "James Robert"}');
```

Functions to Fetch JSON Values

Function	Description
`JSON_TYPE()`	Takes a JSON argument and parses it to give its JSON type.
`JSON_ARRAY()`	Accepts a list of values and returns them as elements of a JSON array.
`JSON_OBJECT()`	Accepts a list of key/value pairs and returns them as pairs in a JSON object.
`JSON_MERGE()`	Accepts more than one JSON document and gives a combined outcome.

Searching and Editing JSON Values

- Following figure extracts the object value for the `name` key from a JSON document:



```
mysql> SELECT JSON_EXTRACT('{"ID": 590, "name": "Lucifer"}', '$.name');
+---------------------------------------------------------+
| JSON_EXTRACT('{"ID": 590, "name": "Lucifer"}', '$.name') |
+---------------------------------------------------------+
| "Lucifer"                                                |
+---------------------------------------------------------+
1 row in set (0.00 sec)
```

Functions Using Path Expressions

Function	Description
`JSON_SET()`	Replaces values for existing paths and adds values for non-existing paths.
`JSON_INSERT()`	Inserts new values without replacing the current ones.
`JSON_REPLACE()`	Replaces the current values and overlooks the new ones.
`JSON_REMOVE()`	Accepts single or multiple paths along with the values to be discarded.

Summary 1-2

- You can upgrade to MySQL 5.7 from MySQL 5.1 either through in-place or logical upgrade.
- Performance Schema is a feature that monitors all server events, stores event details in temporary tables, and holds all system and status variables.
- The sys Schema allows viewing the summary of Performance Schema data.
- MySQL 5.7 provides security enhancements such as policy for automatic password expiry, locking and unlocking accounts, and so on.
- MySQL offers several types of programs such as server startup, installation- or upgrade-based, client, administrative, development, and miscellaneous programs.

Summary 2-2

- Spatial data types and their functions are available only for NDB, InnoDB, MyISAM, and ARCHIVE tables.
- Geometry is the base class for spatial data and has four subclasses namely, Point, Curve, Surface, and GeometryCollection.
- WKT and WKB formats denote geometry objects and are converted into internal format for storing the objects in the database.
- JSON data is stored either as an array or as an object.
- Various MySQL functions are used to obtain JSON values from a JSON document.