

CSIT115 Data Management and Security
CSIT882 Data Management Systems

Architecture of Relational Database Server

Subject Coordinators: Dr Chen Chen, Dr Thanh Le

School of Computing and Information Technology -
University of Wollongong

Architecture of Relational Database Server

Outline

Client-Server Architecture

Basic Operations on Database Server

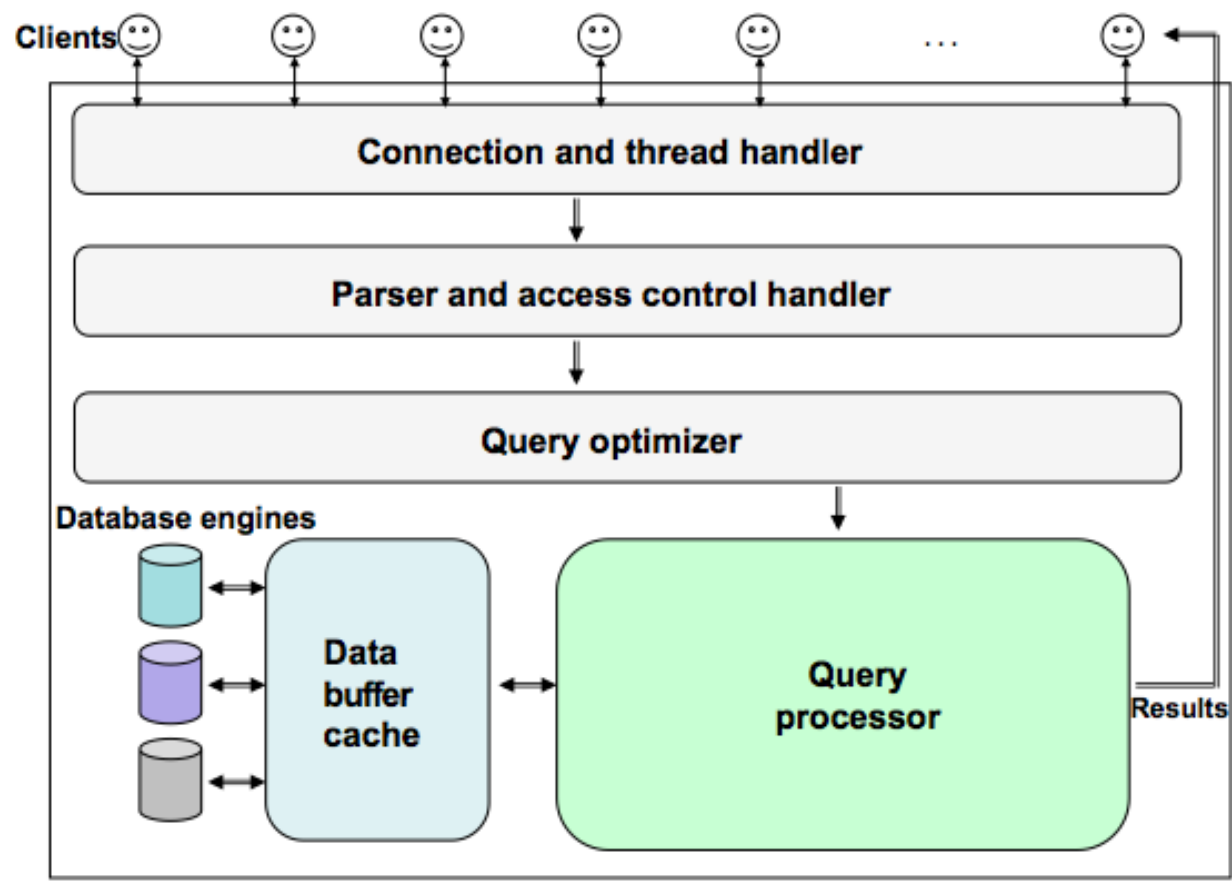
Initialization Variables

Post Installation

Databases

Client-Server Architecture

A **client-server** architecture means, that a number of **clients** connects to a database **server** over a local or wide-area network



In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

3/23

Client-Server Architecture

A database server is implemented over there layers:

- Services not unique to MySQL like: network based client/server tools for connection handling, authentication, security, etc
- DDL and DML processing like query processing, analysis, optimization, caching, and all built-in functions, data entry, data modification, creating database structures, etc
- Database (storage) engines responsible for storing and retrieving data, e.g. [InnoDB](#), [MyISAM](#), [MEMORY](#), [CSV](#), etc

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

4/23

Client-Server Architecture

Client connections are organized in the following way:

- As more than one client can be connected at a time, each client connection gets its own thread within a database server
- A thread resides on one core or on one CPU
- When a client connects to a server then a server authenticates a connection
- Authentication is based on a user name, originating host and password
- Once a client is connected, the server verifies whether it has the privileges to access the relational tables in a database

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

5/23

Architecture of Relational Database Server

Outline

Client-Server Architecture

Basic Operations on Database Server

Initialization Variables

Post Installation

Databases

Basic Operations on Database Server

Starting/stopping a database server

Usually a database server is automatically started at boot-up time of an operating system

A database server can be stopped through a command line interface to an operating system ([shell](#)) for a certain period of time

```
service mysql stop
```

Stopping MySQL server

A database server can be started through a command line interface to an operating system ([shell](#))

```
service mysql start
```

Starting MySQL server

A status of database server can be found through a command line interface to an operating system ([shell](#)) in the following way

```
service mysql status
```

Listing a status of MySQL server

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

7/23

Architecture of Relational Database Server

Outline

Client-Server Architecture

Basic Operations on Database Server

Initialization Variables

Post Installation

Databases

Initialization Variables

At startup time a database server reads the **system initialization variables**

The **system initialization variables** determine the functionality of a database server

For example, see below some of the system initialization variables of MySQL database server

System initialization variables	
Variable_name	Value
auto_increment_increment	1
auto_increment_offset	1
autocommit	ON
automatic_sp_privileges	ON
avoid_temporal_upgrade	
...	...
datadir	/var/lib/mysql/
date_format	%Y-%m-%d
lower_case_table_names	0
max_user_connections	0
...	...

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

9/23

Initialization Variables

The **system initialization variables** are included in the **system configuration files**

To find the locations of **system configuration files** we may process the following commands through command line interface to an operating system

```
mysql --help | grep "Default options" -A 1
```

Listing location of system configuration files

Typically the **configuration files** are located at

```
/etc/my.cnf  
/etc/mysql/my.cnf  
~/.my.cnf
```

Location of system configuration files

In our case a file **mysqld.cnf** with the system initialization variables is located at

```
/etc/mysql/mysql.conf.d/
```

Location of system configuration files

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

10/23

Initialization Variables

The sample contents of system configuration file

```
[mysqld]
user      = mysql
pid-file  = /var/run/mysqld/mysqld.pid
socket    = /var/run/mysqld/mysqld.sock
port      = 3306
basedir   = /usr
datadir   = /var/lib/mysql
tmpdir    = /tmp
lc-messages-dir = /usr/share/mysql
explicit_defaults_for_timestamp
```

System initialization variables

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

11/23

Initialization Variables

To display **ALL** (501) system initialization variables we use **show variables** statement

To display **GLOBAL** (487) system initialization variables we use **show global variables** statement (parameters for new connections)

To display **LOCAL(SESSION)** (487) system initialization parameters we use **show variables** statement (parameters for the current connection)

For example, to find all variables related to updates we use a statement

Listing system initialization variables with 'update' in their names

```
show variables like '%update%';
```

For example, to find a value of variable **lower_case_table_names** we use a statement

Listing system initialization variables whose name starts from 'lower_case_table'

```
show variables like 'lower_case_table%'
```

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

12/23

Initialization Variables

To change a value of dynamic system initialization variables we use `set` statement

For example to change a value of system initialization variable `sql_safe_updates` to 0 we use the following `set` statement

```
set sql_safe_updates=0
```

Changing a value of system initialization variable

Some of the system initialization variables are not dynamic and it cannot be changed with `set` !

For example, a variable `lower_case_table_names` is not dynamic and it cannot be changed with `set`

```
set lower_case_table_names=1
```

Changing a value of system initialization variable

Error message

```
ERROR 1238 (HY000): Variable 'lower_case_table_names' is a read only variable
```

The system initialization variables, that are not dynamic must be changed in a **system configuration file** (stop server, change variable, start server)

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

13/23

Architecture of Relational Database Server

Outline

Client-Server Architecture

Basic Operations on Database Server

Initialization Variables

Post Installation

Databases

Post Installation

Just after installation there is only one user **root** available on the installed system

First we start **mysql** client and we connect as a user **root** without a password !

```
mysql -u root
```

The first start of mysql client

As it is an evident security risk we must set a password for **root** user

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'password';
```

Changing password

To find what other users can connect to the system execute a statement:

```
SELECT user, host, HEX(authentication_string) FROM mysql.user;
```

Finding all users that can connect to the system

Sample listing of user names

user	host	HEX(authentication_string)
root	localhost	2A383146354532314533353430374438386426432...30394
mysql.sys	localhost	2A5448495349534E4F544156414C4944504155345...44484
csit115	localhost	2A323045343946324432303337373739383133336...44433

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

15/23

Post Installation

After a password to a user `root` is changed, any future connection as a user `root` must be performed as follows

```
mysql -u root -p -v -c
```

Sample connection as a user root

Finally, we must remember, that MySQL `root` user is completely different from an operating system user `root` !

To find what databases are available on the system we use a statement

```
show databases;
```

Listing databases

A database `information_schema` is commonly called as a data dictionary and it contains information about the relational tables, columns, constraints, etc

To list the names included in a database `information_schema` we make the database a default one

```
use information_schema;
```

Making 'information_schema' database a default one

[In HTML view press 'p' to see the lecture notes](#)

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

16/23

Post Installation

And, then we simply "say"

```
show tables;
```

Listing table names

Now, we know the names of the relational tables in `information_schema` database and it is possible to access the data dictionary tables

For example, we access a relational table `user_privileges` to find what privileges are granted to the users

```
SELECT * FROM user_privileges;
```

Listing user privileges

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

17/23

Post Installation

The privileges granted to the users `root` and `csit115` are the following

User privileges			
GRANTEE	TABLE_CATALOG	PRIVILEGE_TYPE	IS_GRANTABLE
'root'@'localhost'	def	SELECT	YES
'root'@'localhost'	def	INSERT	YES
'root'@'localhost'	def	UPDATE	YES
'root'@'localhost'	def	DELETE	YES
'root'@'localhost'	def	CREATE	YES
'root'@'localhost'	def	DROP	YES
...
'csit115'@'localhost'	def	SELECT	NO
'csit115'@'localhost'	def	INSERT	NO
'csit115'@'localhost'	def	UPDATE	NO
'csit115'@'localhost'	def	DELETE	NO
...

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

18/23

Architecture of Relational Database Server

Client-Server Architecture

Basic Operations on Database Server

Initialization Variables

Post Installation

Databases

Databases

In order to create the relational tables we have to create a **database** first
While connected as a user **root** we process a statement

```
CREATE DATABASE database-name;
```

Creating a database

We can always drop it with

```
DROP DATABASE database-name;
```

Dropping a database

To be able to create a database a user must have **CREATE DATABASE** privilege and to drop a database a user must have **DROP DATABASE** privileges

To list all created databases we process a statement

```
show databases;
```

Listing available databases

To make a database a default one, we process a statement

```
use database-name;
```

Setting a default database

[In HTML view press 'p' to see the lecture notes](#)

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

20/23

Databases

For example, to use a database `csit115` we process a statement

```
use csit115;
```

Setting 'csit115' database as a default database

Then, if a relational table `DEPARTMENT` is created in a database `csit115` then we can access the table with a simple

```
SELECT * FROM DEPARTMENT;
```

SELECT statement

If a relational table `COURSE` is created in a database `university` then we can access it either through

```
use university;  
SELECT * FROM COURSE;
```

Setting university database as a default database

or through prefixing a table name with a database name

```
SELECT * FROM university.COURSE;
```

Accessing a relational table COURSE in a database university

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

21/23

Databases

Only one database can be a **default on** at a time, after processing of **use** statement

A user can access many databases at a time through prefixing the names of relational tables located in the other databases with an appropriate **database name**

A database can be dropped with

```
DROP DATABASE database-name;
```

Dropping a database

In HTML view press 'p' to see the lecture notes

[TOP](#)

Created by Janusz R. Getta, CSIT115 Data Management and Security, Autumn 2023

22/23

References

Cabral S., Murphy K., MySQL Administrator's Bible, Wiley Publications, 2009 (Available online through UOW Library)

[How to ... ? Cookbook, How to manage MySQL database server ? Recipes 8.1, 8.2, 8.3, and 8.4](#)