

MySQL



Tools, Syntax, Create, Insert, and Select

COS216

AVINASH SINGH

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF PRETORIA

DATABASE – OVERVIEW

- Database Management System (DBMS)
 - Set of tools to create and maintain a database
- Data stored in simple files
- Allows advanced and fast access to data
 - Efficient searching and sorting algorithms
 - Mutual exclusion and locking
 - Atomic operations
 - Many more features

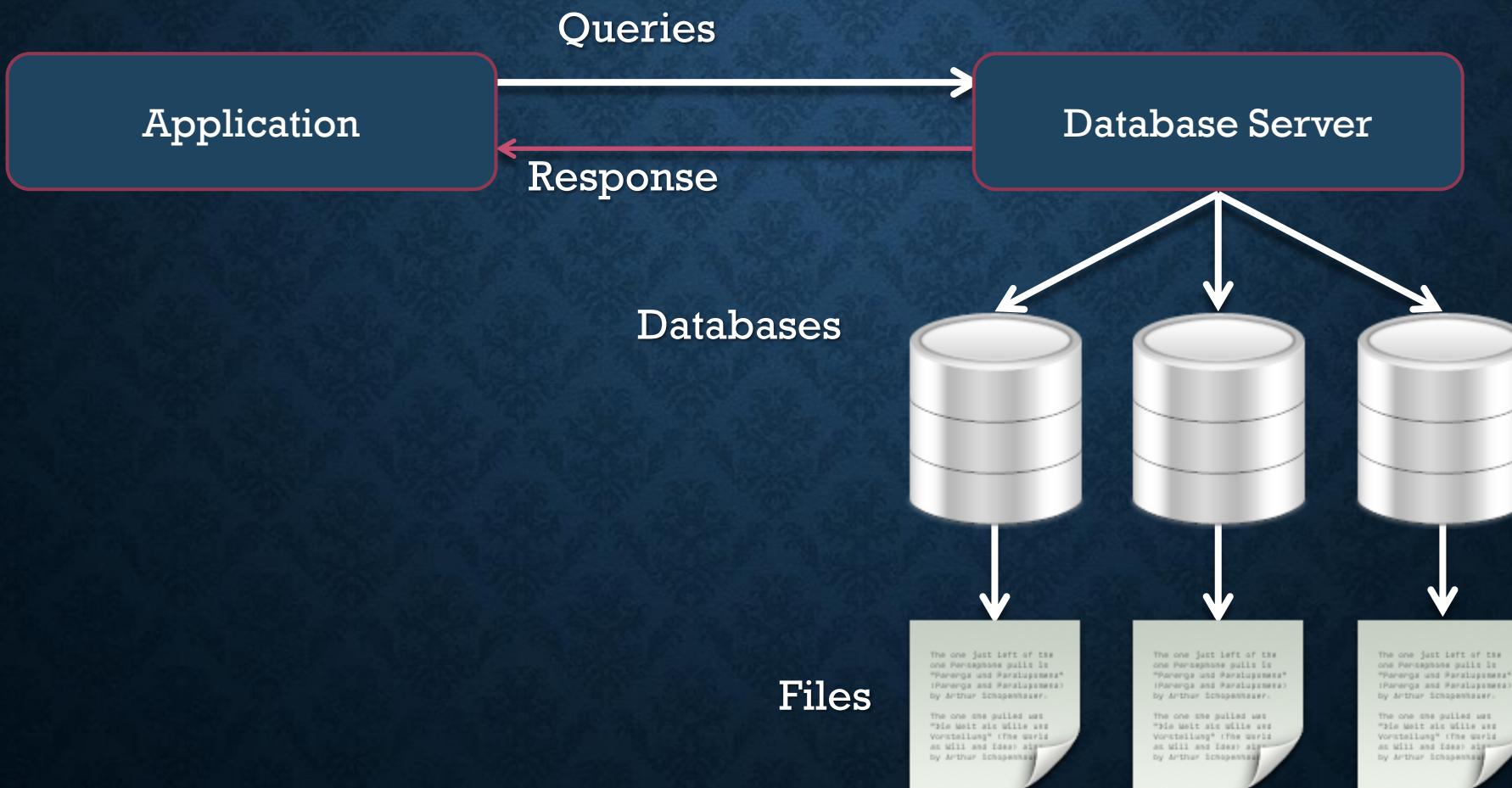
DATABASE – OVERVIEW

- Hierarchical DBMS
 - A tree with the data as nodes
 - Developed by IBM in 1968
 - Windows Registry is a hierarchical DBMS
- Network DBMS
 - Allows many-to-many relationships
 - Fortran, Cobol, Pascal and C++ implemented sets/records like this
- Relational DBMS (RDBMS)
 - Is a network DBMS with relationships treated as tables

DATABASE – OVERVIEW

- Object-Oriented DBMS
 - Focus is on object-oriented design
 - Can handle many new data types, audio, video, images
 - More expensive than a traditional DBMS
- NoSQL
 - Around since the 1960s but only gained popularity recently with Google, Facebook, etc
 - Tabular relationships are replaced with key-values, wide columns, graphs, and documents
 - Some operations are faster, more scalable, has a simpler design, and can be easily used on clusters

RDBMS – ARCHITECTURE



RDBMS – TABLES

Columns (Fields)

Rows (Entries)

ID	Name	Gender	Age

RDBMS – RELATIONSHIPS

User Table

ID	Name	Address

Address Table

ID	Number	Street



RDBMS – QUERIES



The DBMS takes instructions from the user



The instructions are codified using a language



SQL is one such language



Alternatives to SQL include 4D QL, .QL, Datalog, OttoQL, ISBL, and XQuery

SQL – OVERVIEW

- Structured Query Language (SQL)
- Special-purpose language used to access data from a relational database
- Standardised in 1986 (ANSI) and 1987 (ISO)
 - Although standardised, it might work slightly different amongst DBMSs
- SQL has benefits over older query languages
 - Many records can be retrieved with a single command
 - Records can be accessed without an index
- Mathematical based
 - Relational algebra
 - Tuple relational calculus

SQL – CRUD OPERATIONS

- A SQL should at least be able to:
 - **Create**
 - **Read**
 - **Update**
 - **Delete**



SQL - DATABASES

MySQL

PostgreSQL

SQLite

Microsoft SQL

Microsoft Access

Oracle SQL

IBM DB2

SAP Sybase

Many more ...

MYSQL – OVERVIEW

- Most widely used RDBMS on the internet
- Was free and open-source under GPLv2 or commercial licenses
- Named after co-founders daughter: My
- Pronunciations:
 - My – SQL
 - My – Sequel



MARIADB – OVERVIEW

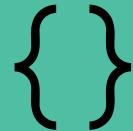
- MySQL was originally maintained by Sun Microsystems
 - Company behind Java, Netbeans, PostgreSQL, Intel Xeon, AMD Opteron, OpenOffice
 - Sun Microsystems was a huge proponent of free and open-source software
- Oracle (3rd largest software company) purchased Sun Microsystems in 2009
 - Oracle is profit-driven and slowly turns Sun Microsystem software into paid commercial products
 - MySQL vs MariaDB, OpenOffice (handed over to Apache) vs LibreOffice



MARIADB – OVERVIEW

- A few original MySQL developers have forked MySQL into MariaDB
 - MariaDB is intended to always stay free and open-source
 - Has the exact same syntax as MySQL
 - Most web hosting companies are busy or have already switched over to MariaDB
 - XAMPP also now uses MariaDB
- Named after the main developer's daughter: Maria

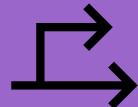
MySQL – OVERVIEW



MySQL and MariaDB are written in C/C++



Cross-language, cross-platform, cross-architecture



MySQL released in 1995,
MariaDB forked in 2009

MYSQL – USAGE

- Many applications use MySQL/MariaDB
 - WordPress
 - Joomla
 - phpBB
 - MyBB
 - TYPO3
 - MODx
 - Many more ...

MySQL – USAGE

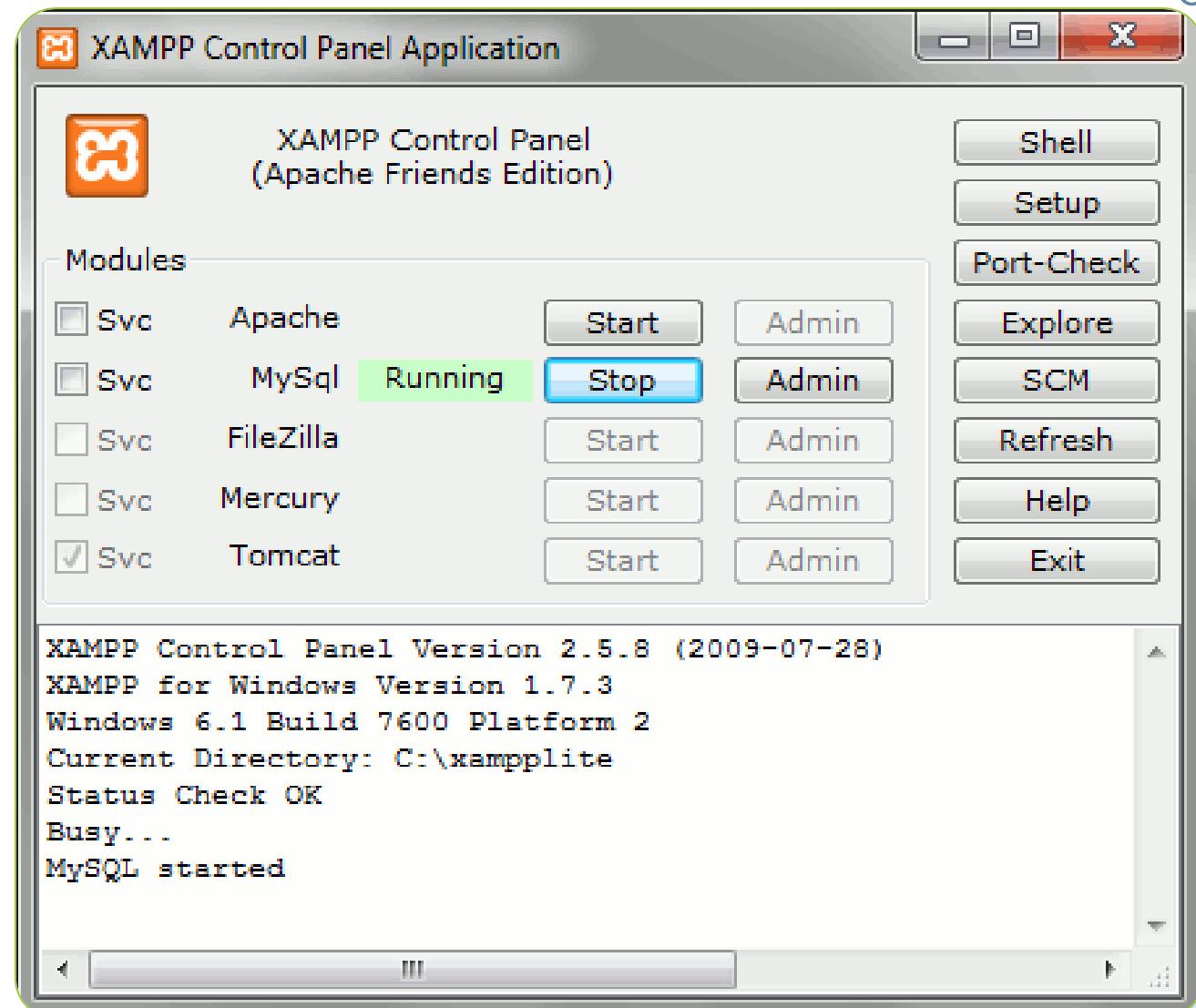
- Many companies use MySQL/MariaDB
 - Google
 - Wikipedia
 - Facebook
 - Twitter
 - Flickr
 - YouTube
 - Many more ...

MYSQL – TOOLS

- MySQL has two main programs
- Server
 - The server that hosts the databases
 - Receives queries from clients
- Client
 - Terminal or graphical interface
 - Send queries to server and display the results

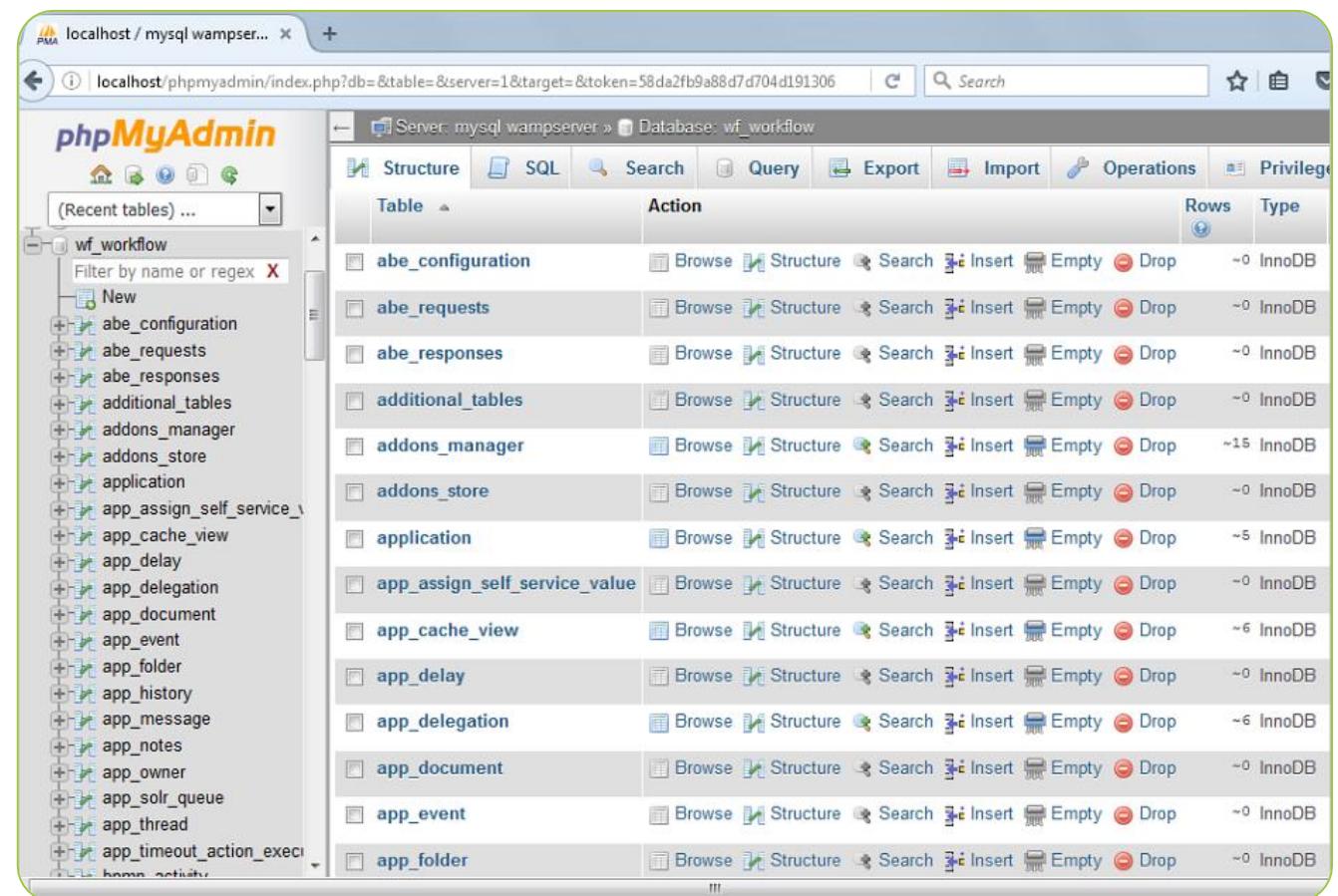
MYSQL – TOOLS

- MySQL/MariaDB is part of the web application software stacks
 - XAMPP
 - LAMP
 - WAMP
 - MAMP



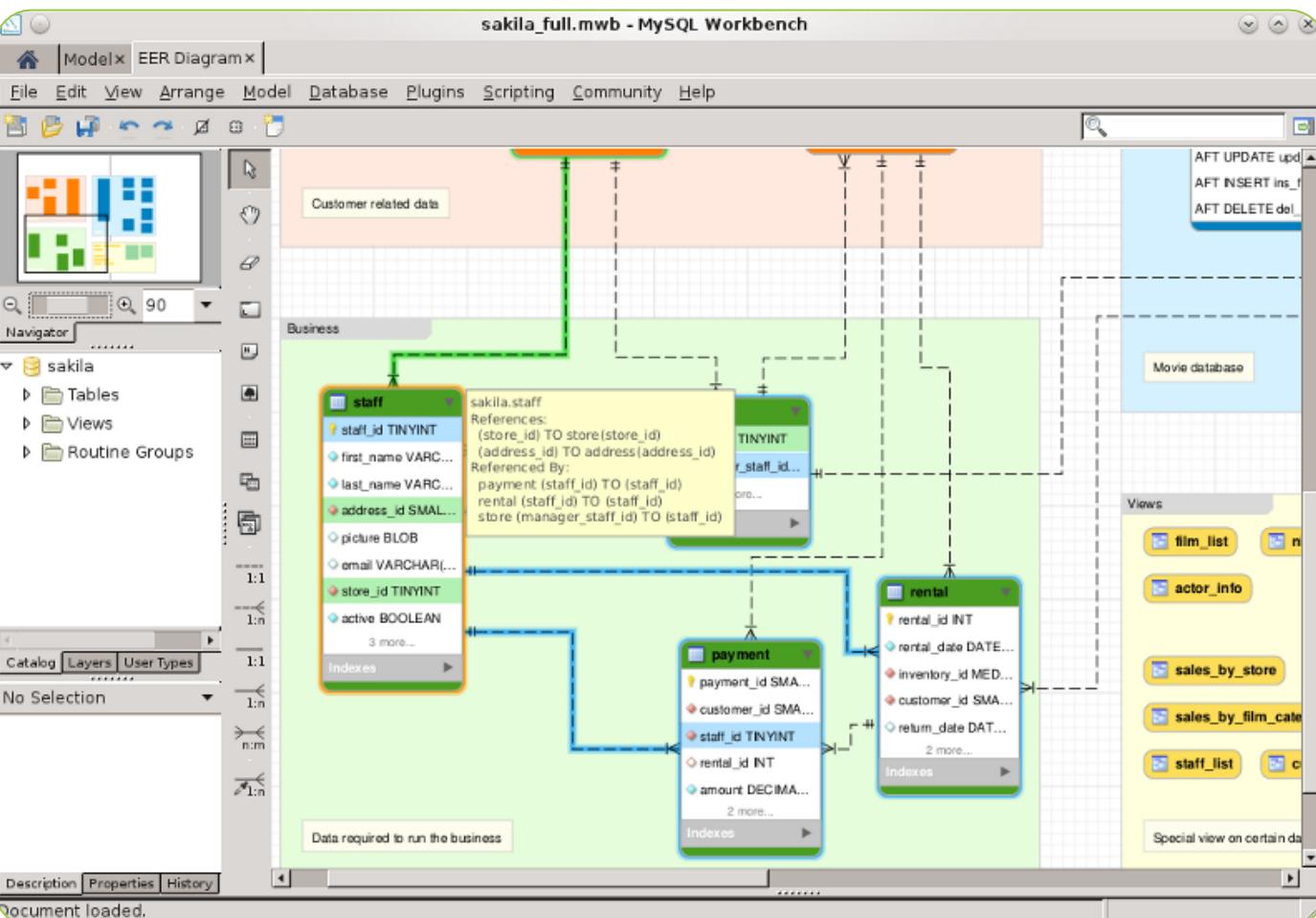
MYSQL – TOOLS

- Web application software stacks often come with phpMyAdmin
- Most commercial web hosts come with phpMyAdmin
- Graphical web interface to MySQL
- Also available in XAMPP



MYSQL – TOOLS

- Graphical interface to MySQL



MYSQL – TOOLS

- Windows:
 - Download the installer from the MySQL website and follow the instructions.
- Debian, Ubuntu, LinuxMint:
 - Client: `sudo apt-get install mariadb-client`
 - Server: `sudo apt-get install mariadb-server`
 - Workbench: `sudo apt-get install mysql-workbench`

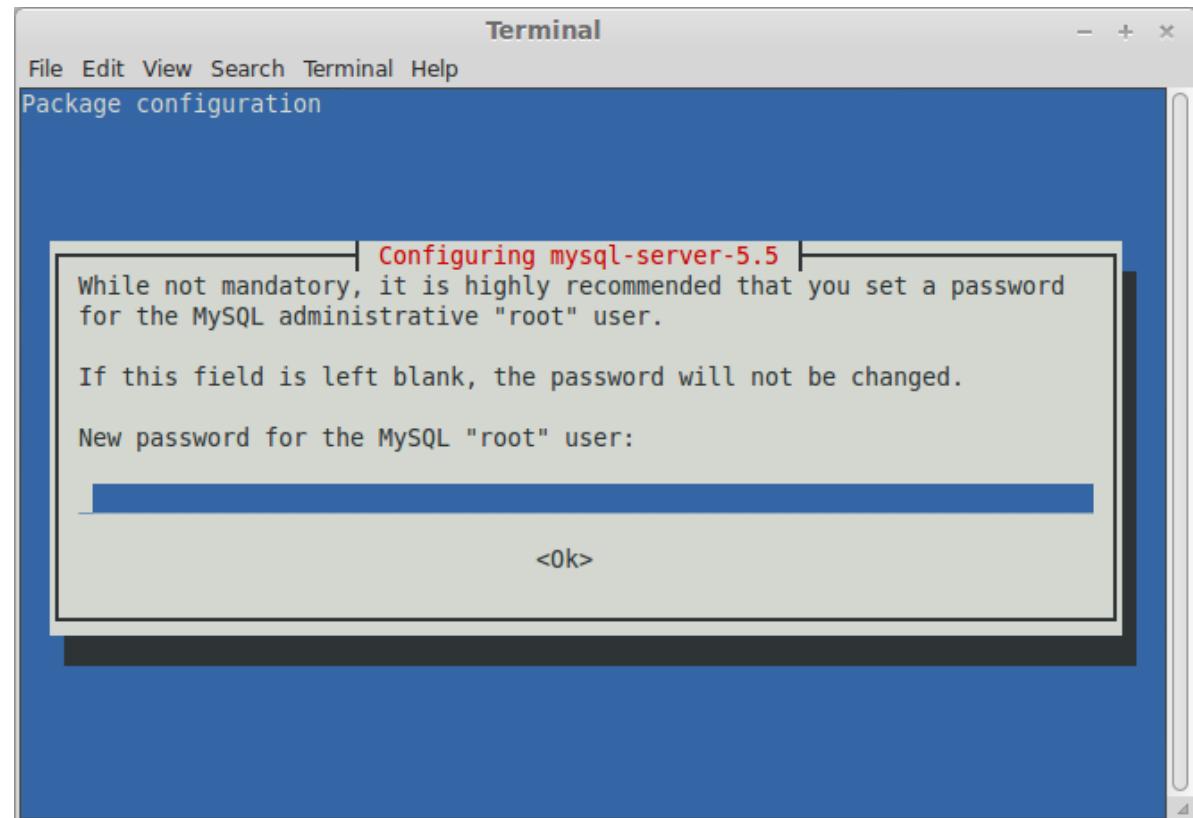
MYSQL - TOOLS

- Install the server:

```
sudo apt-get install mariadb-server
```

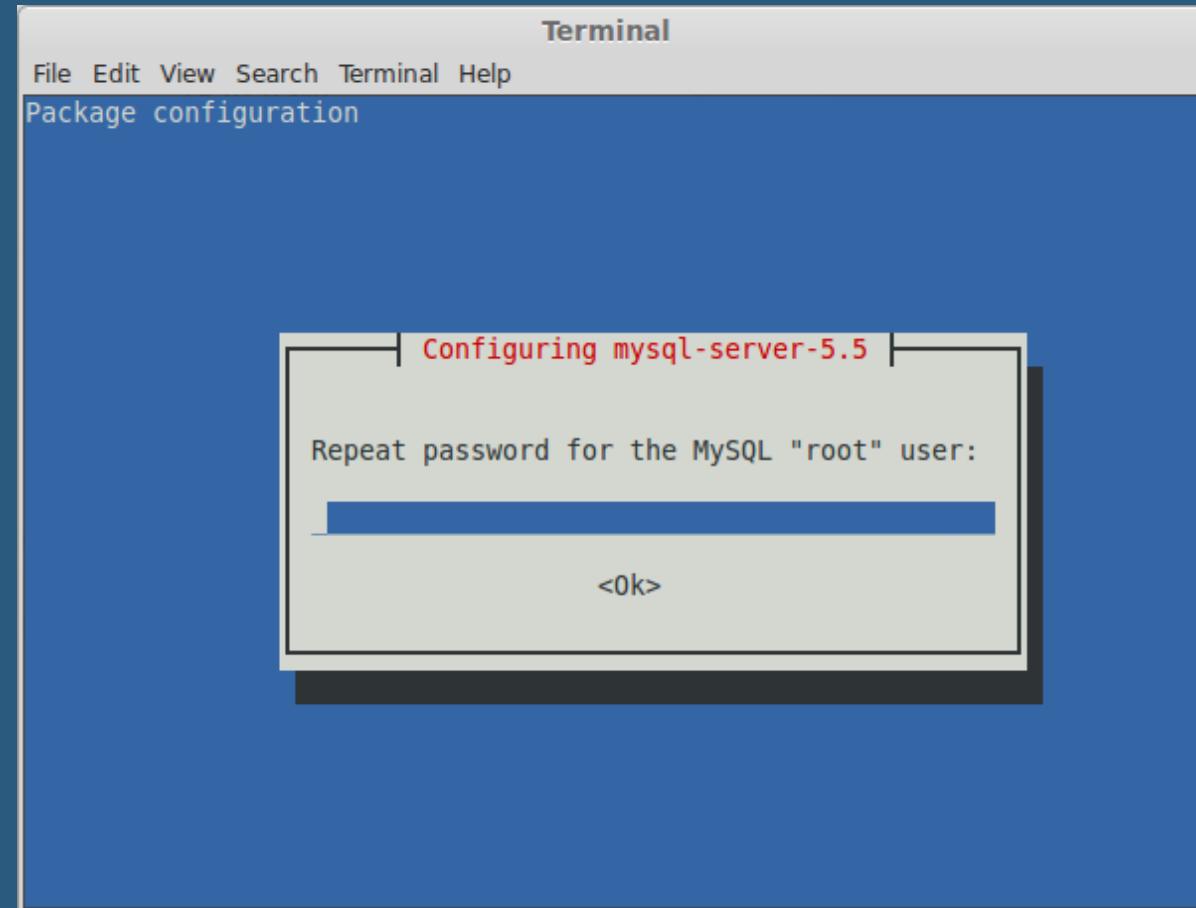
MYSQL – TOOLS

- Enter the admin password for the server
- Used for the root (super user) of the database



MYSQL – TOOLS

- Repeat the root password



MYSQL - TOOLS

- Install the client

```
sudo apt-get install mariadb-client
```

MYSQL – CONNECTION

- Open the terminal and connect to the MySQL server

```
mysql --host <host> --port <port> --user <user> --pass <pass> <database>
```

- Or short-hand version

```
mysql -h <host> -P <port> -u <user> -p <pass> <database>
```

MYSQL – TOOLS

- The default user is **root**
- The default port is **3306**
- The password is often not allowed as part of the statement (just add **-p** flag)

```
mysql -h localhost -u root -p
```

- You will then be asked to enter the password (note that password do not show when typed in Linux terminals)

MYSQL – TOOLS

- List all available databases:

```
show databases;
```

- Create a new databases:

```
create database <dbname>;
```

- Delete a databases:

```
drop database <dbname>;
```

- Use/access a databases:

```
use <dbname>;
```

MYSQL – TOOLS

- Create a new user with advanced options:

```
create user '<username>'@'<host>' identified by '<password>';
```

- Create and grant all privileges to user:

```
grant all privileges on <dbname>.* to '<username>'@'<host>'  
identified by '<password>';
```

- Delete a user:

```
drop user '<username>'@'<host>';
```

MYSQL – TOOLS

- Once a database was selected, list all the tables:

```
show tables;
```

- Import an existing script/file (eg: backup.sql):

```
source <filename>;
```

MYSQL – WHEATLEY

- Use your text found in db_password
(wheatley.cs.up.ac.za/uXXXXXXXXXX/db_password) to login to mysql
- Remember that the database is shared with COS221, don't accidentally delete your databases

```
mysql -h wheatley.cs.up.ac.za -u <CS username> -p  
enter <db_password>  
show databases;  
use <CS username>;  
<do something with the database>  
exit;
```

MYSQL – COMMENTS

```
/* C-style comments */  
-- Standard SQL comments --
```

MYSQL – DATA TYPES



MySQL has 3 classes of data types



Numbers

Integers
Decimals



Strings

Text
Blob



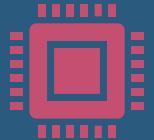
Times

Date
Time

MYSQL – DATA TYPES



Note that MySQL does not natively support some types



Booleans

Use an integer instead
TINYINT is the smallest integer with only 1 byte



Unix timestamps

Use an integer instead



MySQL trims leading 0s from numbers

Number that can start with a 0 (eg: phone or ID numbers) should be stored as strings, not integers

MYSQL – INTEGERS

- All integers can be signed or unsigned
- Unsigned integers have twice the upper bound (maximum)

Type Name	Bytes	Minimum	Maximum
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INTEGER / INT	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807

MYSQL – DECIMALS

- Decimal types can have up to 2 parameters
 - Display length: the total number of digits
 - Decimal length: the total number of decimal digits
 - Example: FLOAT(5,2) has a range from -999.99 to 999.99

Type Name	Bytes	Precision	Comment
FLOAT	4 or 8	24 places	Small decimal value
DOUBLE / REAL	8	53 places	Large decimal value
DECIMAL / NUMERIC	8	65 places	A double stored as a string, allowing a fixed decimal point

MYSQL - TEXT

Type Name	Maximum Length	Bytes	Comment
CHAR	255 to 65535	Depends	Fixed length 255 char limit in old versions, up to 65535 in new versions, depending on encoding
VARCHAR	255 to 65535	Depends	Variable length 255 char limit in old versions, up to 65535 in new versions, depending on encoding
TINYTEXT	255	256B	
TEXT	65535	64KB	
MEDIUMTEXT	16777215	16MB	
LONGTEXT	4294967295	4GB	

MYSQL – TEXT

- TEXT is used to store raw text of variable length
- CHAR is used to store short fixed length strings, takes a length parameter
 - Example: SA ID numbers always have a fixed length
 - CHAR(13) – exactly 13 characters are stored
- VARCHAR is used to store short variable length strings, takes a length parameter
 - Example: Names are short, but have variable lengths
 - VARCHAR(128) - maximum length of 128 characters, but any length from [0,128] allowed

MYSQL – BLOBS

- BLOB is used to store binary data, such as images or PDFs
- Larger binary data should be stored as normal files inside a directory on the server, and the database should only contain a file path

Type Name	Maximum Length	Bytes
TINYBLOB	255	256B
BLOB	65535	64KB
MEDIUMBLOB	16777215	16MB
LONGBLOB	4294967295	4GB

MYSQL – TIMES

- YEAR takes a parameter as either 2 or 4
- Note that MySQL's timestamp is not the Unix timestamp format
 - Use INT to store Unix timestamps

Type Name	FORMAT
TIME	HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
YEAR	YY or YYYY
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS

MYSQL – CREATE

- Create a new table

```
CREATE TABLE student
(
    id CHAR(13) PRIMARY KEY,
    firstname VARCHAR(200),
    surname VARCHAR(150),
    birthday DATE
);
```

- Note that statements must end with a semicolon (;)
- Note that the last statement inside the CREATE (birthday DATE) cannot end with a comma, otherwise MySQL will throw an error

MYSQL – CREATE

```
CREATE TABLE address
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    street VARCHAR(200) NOT NULL,
    city VARCHAR(100) NOT NULL,
    zipcode CHAR(4)
);
```

- NOT NULL: must have a value, if not specified, the value can be NULL
- PRIMARY KEY: a unique key in the table that identifies the entry/row
- AUTO_INCREMENT: instead of manually specifying a primary key, let MySQL manage the key for you. MySQL will start with a key of 1, and every new row inserted will have an incremented key (1, 2, 3, 4, 5, ...)

MYSQL – CREATE

```
CREATE TABLE student
(
    id CHAR(13) PRIMARY KEY,
    firstname VARCHAR(200),
    surname VARCHAR(150),
    birthday DATE,
    address INT NOT NULL,
    FOREIGN KEY (address) REFERENCES address(id)
);
```

- FOREIGN KEY: create a relationship between two tables. Link an attribute in one table (student.address) to the primary key of another table (address.id)

MYSQL – INSERT

```
INSERT INTO address (street, city, zipcode)  
VALUES ("23 Boom Str", "Pretoria", "0001");
```

```
INSERT INTO student (id, firstname, lastname, birthday, address)  
VALUES ("9401234052087", "Satoshi", "Nakamoto", "1994-01-23", LAST_INSERT_ID());
```

- If the primary key is auto incremented, do not provide a key, MySQL will do it for you. If it is not auto incremented, you have to provide the key manually
- Dates are provided in quotes, various formats are possible
- The address must be inserted before the student, since the student table has a foreign key to the address table
- LAST_INSERT_ID is a function that retrieves the last inserted auto incremented key in the entire database. Easily retrieve the ID from the previous insert statement to use as a foreign key in the next insert statement

MYSQL – SELECT

- Retrieve rows or values from tables

```
SELECT * FROM student;
```

```
SELECT lastname, firstname FROM student WHERE id = "9401234052087";
```

- Either all columns are retrieved (*), or only specific columns
- Select all rows from a table, or only return rows that adhere to specific selection parameters (WHERE)

MYSQL – SELECT

```
SELECT * FROM student  
WHERE birthday > "1990-01-01" AND firstname LIKE "%tosh%";
```

- Various Boolean expressions are possible (AND, OR, equal =, not equal <>, <, >, <=, >=). The != operator works as well, but not in older versions (use <>)
- The LIKE operator uses pattern matching where % means “anything”
- This example query returns all students born in or after 1990 and contain the string “tosh” in the first name

MYSQL – SELECT

```
SELECT * FROM student ORDER BY surname DESC LIMIT 5;
```

- Order the returned rows in ascending (ASC) or descending (DESC) order according to a specific column
- Limit the maximum number of rows to return
- This example query returns a maximum number of 5 students sorted according to their surname (from Z to A)

MYSQL – SELECT

```
SELECT surname, COUNT(*) AS total FROM student GROUP BY surname;
```

- Count the total number of rows returned by a query
- Other functions are available (MIN, MAX, AVG, SUM, STD, etc ...)
- Give a temporary alias to a variable using AS
- Group the returned rows according to a specific column
- The example query returns all distinct surnames and the corresponding number of students who have that surname (eg: Zuma 42, Ramaphosa 3, Malema 6)



MYSQL