INFO 151
Web Systems and Services

Week 9 (T1)

Dr Philip Moore

Dr Zhili Zhao

Course Overview

Weeks 1 – 3

- Introduction to Web Systems and Services
- Creating Web-Pages and Web-Sites with a Markup Language
- Introductory HTML 4 and HTML 5 with CSS

Weeks 4 – 6

- Client-Side Web Programming
- Object Oriented Programming
- Introductory and further JavaScript

Weeks 7 – 9

- Server-side Programming
- Introductory PHP
- Introduction to Database, SQL, and MySQL

Overview

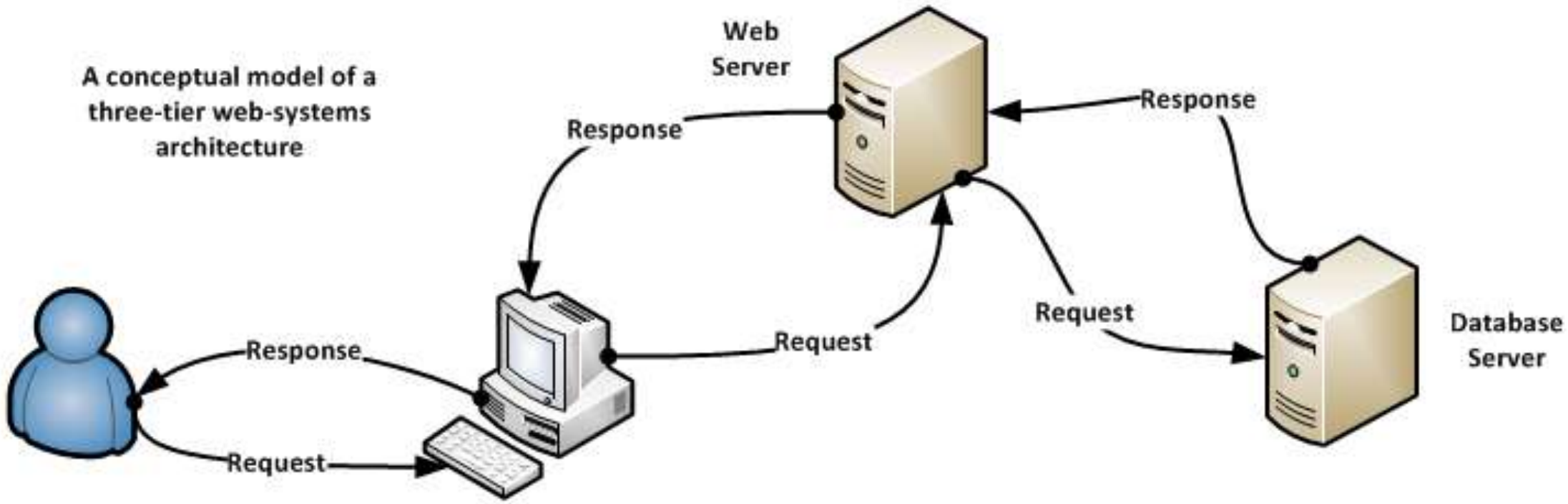
- In this week's tutorials we will introduce the required software to create and run PHP scripts and MySQL and:
  - Review web-based systems introduced in week #1
  - Introduce practical 'real-world' use of a database and briefly consider security for database applications
  - Introduce the database basics and the Structured Query Language (SQL) with worked examples
  - Show how to create, manage, and access a database in MySQL server within the NetBeans environment including the use of a PHP script

Review
Database Technologies
Web Systems Architectures

Web Systems Database Applications

- In developing web-systems database applications
  - In the client tier
    - Clients computer systems
    - The web browsers
- The client tier connects to the middle tier
  - In the middle tier
    - The web server
    - The scripting engine
    - The scripts (PHP)
- The middle tier connects to the database server

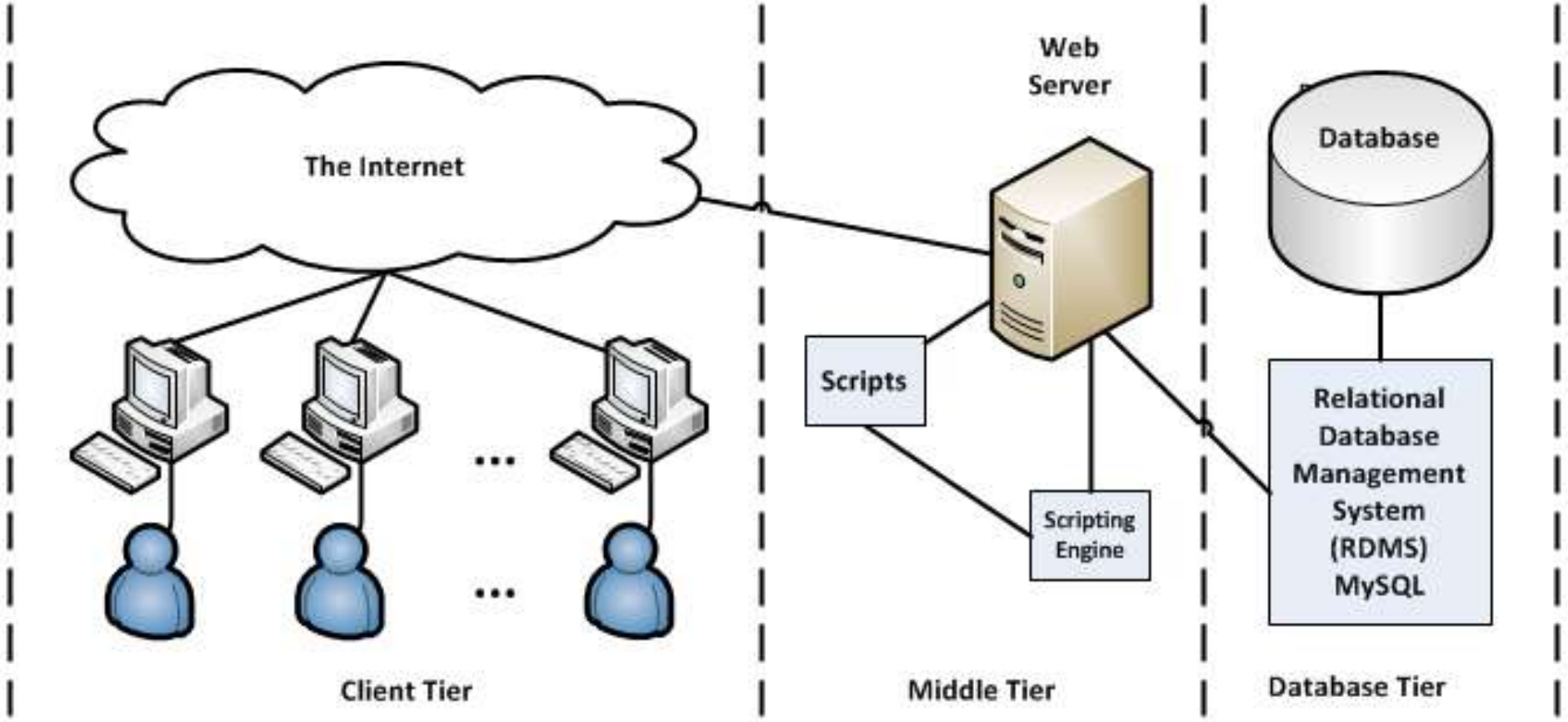
Three-Tier Web-Systems Architecture



Web System Architectures

- The three web-systems architectures are design models of typical architectures
- In *'real-world'* web systems
  - The configuration of physical servers will vary
  - There are physical servers in the second and third layers
  - There is a separate database server (Holding for example the MySQL server)
  - All the physical components are created within a single physical server using dedicated partitions (*virtual servers*)
- The design of a web system is based on
  - The size of a web-site measured in terms of the anticipated number of 'hits'
  - The database function will be designed based on the number of user records

Three-Tier database Application Architecture

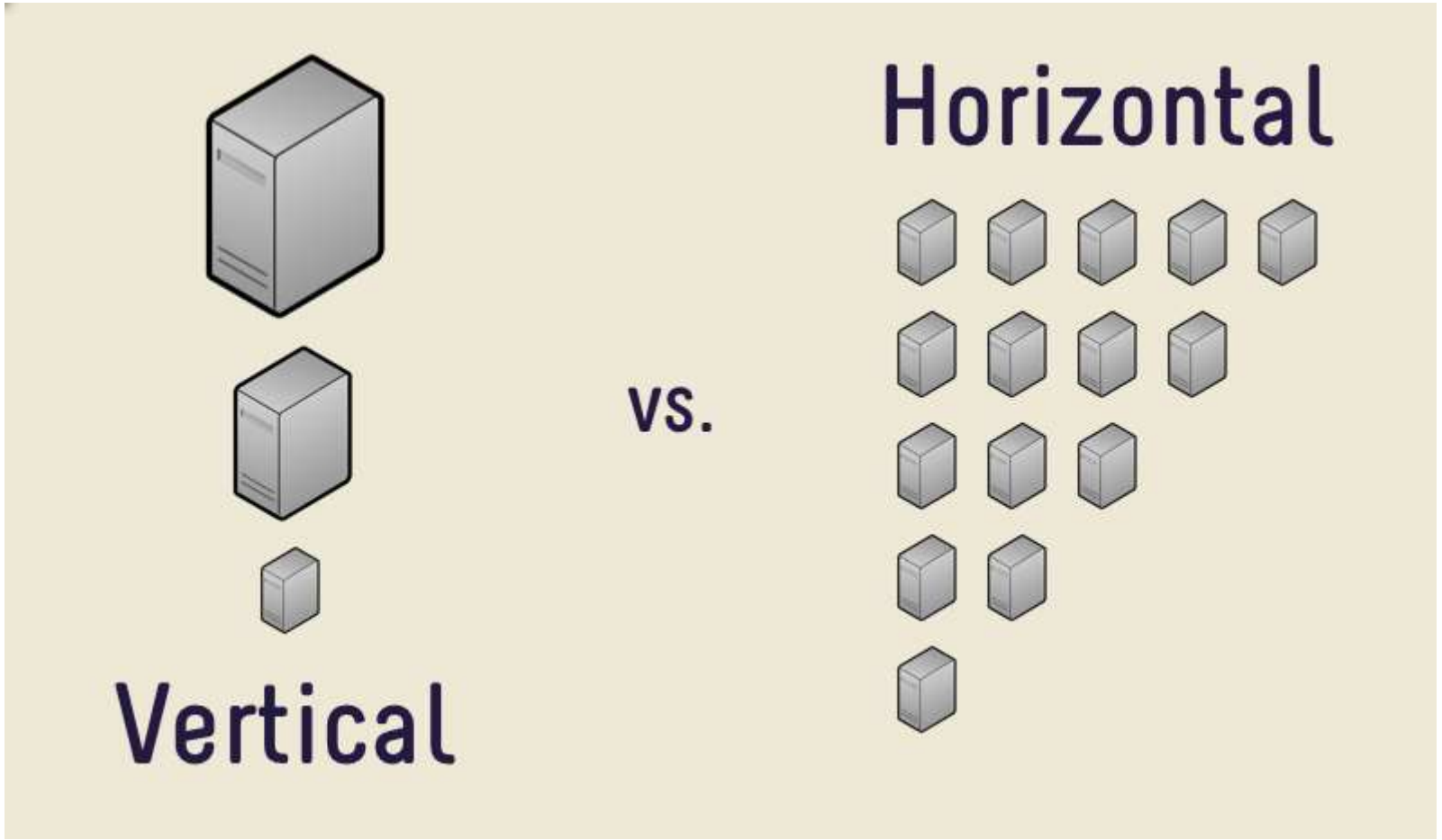


A three-tier architectural model of a web-systems database application

Database Technologies

- There are a number of database technologies in *'real-world'* Internet systems
  - *Relational Database Management Systems* (RDMS) (*structured* data)
  - *NoSQL* database systems (*unstructured* data)
- From a web systems design perspective
  - A RDMS database is located and run from a single server (or *virtual server*)
  - A NoSQL database is generally implemented using *horizontal* and *vertical* scaling (not introduced in this course)

Server Scaling



Web Systems Database Technologies

- Web systems applications (both large and small) generally use cloud-based (Internet) systems
  - A typical example of a cloud-based system is an email application
- Current large web-systems applications often use NoSQL database systems where:
  - Data may be stored in multiple locations within a country (or) in multiple locations in many countries
  - The data storage uses both *horizontal* and *vertical* scaling
  - MySQL is generally restricted to small (local) scale applications
- This course is restricted to the use of the two and three tier architecture using MySQL server

Basic Database Concepts

Database Basics

- A database
  - Is an organized collection of data (stored and accessed electronically)
- A database-management system (DBMS) Is a software application that enables
  - Interaction between a database and users (including other web-systems)
- A general-purpose DBMS enables
  - The definition, creation, querying, update, and administration of databases
- A database is generally stored in a DBMS-specific format which is not portable and typically uses SQL

Relational Database Management Systems

- A relational database management systems (RDMS)
  - Is a DMS based on the *relational model* (RM) which in turn is based on *relational calculus* (addressed in another course)
  - The RM is an approach to managing data using a structure and language consistent with *first-order predicate logic*
  - In a RM all data is represented in terms of *tuples* which are *grouped into relations*
- RDMS operate using the structured query language (SQL)
- RDMS has a specific terminology

RDMS Terminology

- There is a specific terminology used for databases:
  - *Database*: a repository to store data
  - *Table*: a part of a database that stores data related to an entity – e.g., a customer in an on-line web-site
  - *Attribute*: for the columns in a table – all rows have the same attributes
  - *Row*: (a data record) contains values for each attribute
  - *Relational model*: a formal model that uses the database / tables / attributes to store and manage data and their relationships
  - *Relational Calculus*: is a non-procedural query language using mathematical predicate calculus (instead of algebra) and it provides:
    - The description about the *query* to get the *result* (whereas) relational algebra provides the *method* to get the *result*

RDMS Terminology

- There is a specific terminology used for databases:
  - *RDMS*: a software application to manage date in a database based on the RM
    - (MySQL server is a RDMS
  - *SQL*: (see later slide)
  - *Constraints*: restrictions or limitations on tables and attributes)
  - *Primary key*: an attribute (a row) is a unique identifier for a table
  - *Index*: a data structure used for fast access to rows in a table)
  - *Entity-relationship (ER) modelling*
    - This term relates to a technique used to describe 'real-world' data (a conceptual model) in terms of entities / attributes / relationships

What is a Tuple?

- In the context of a relational database:
  - A tuple is one record (or one row in a database table)
- The information in a database:
  - Can be thought of as a spreadsheet
  - With columns (known as fields or attributes) representing different types of information
  - For example: a person table may have data such as:
    - `first_Name, surname, Initials, email, home telephone, work, telephone, etc`
- A tuple (representing all the information from each field) are associated with a single record

Database Records

Winery ID	Winery name	Address	Region ID
1	Moss Brothers	Smith Road	3
2	Hardy Brothers	Jones Street	1
3	Penfolds	Artherton Road	1
4	Lindermans	Smith Avenue	2
5	Orlando	Jones Street	1

Database Tables

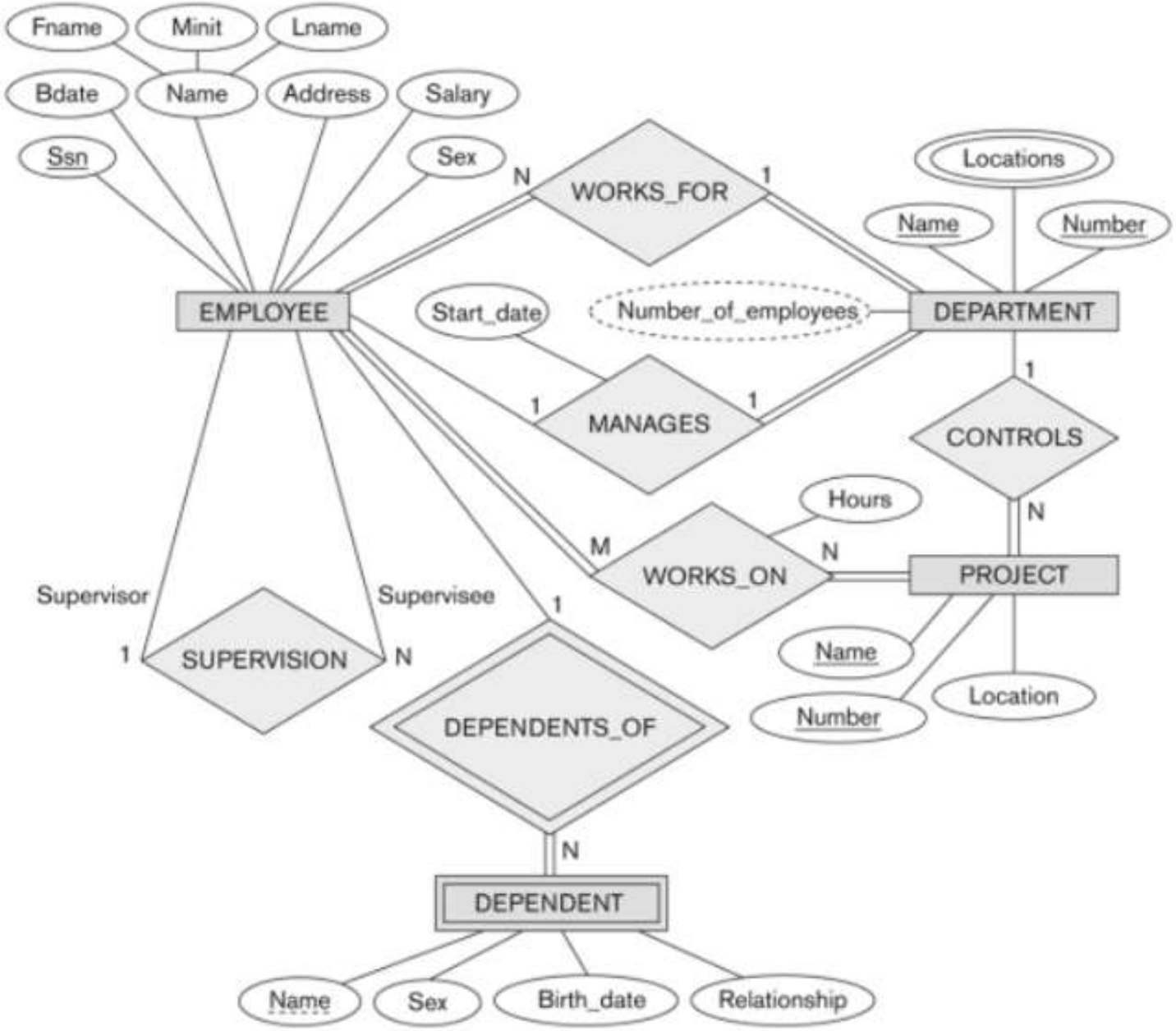
- Shown are two related database tables
  - The *Winery Table*
  - The *Region Table*
- The two tables form part of a much larger database
- Each table will have a unique identifier (*primary key*)
  - In the *Winery Table* it may be the *Winery ID*
  - In the *Region Table* it may be the *Region ID*

Winery ID	Winery name	Address	Region ID
1	Moss Brothers	Smith Road	3
2	Hardy Brothers	Jones Street	1
3	Penfolds	Artherton Road	1
4	Lindermans	Smith Avenue	2
5	Orlando	Jones Street	1

Region ID	Region name	State
1	Barossa Valley	South Australia
2	Yarra Valley	Victoria
3	Margaret River	Western Australia

Entity Relationship Diagram

- An Entity Relationship Diagram (ERD)
- From the diagram we can see relationships between:
  - Employees
  - Managers
  - Departments
  - Dependencies
  - Projects
  - etc



Normalisation

Normalisation

- The process of normalization:
  - Is designed to remove duplication of data records in a database
  - Record duplication (of records and data) causes issues in maintaining and updating a database with potential errors
- Normalisation follows the principles of relational calculus
  - Using the Structured Query Language (SQL) which we address in this tutorial
- A correctly designed database is created from the ER model with the relationships recreated in the database

Normalisation

- In normalization:
  - The initial data is generally set in a single table
  - This is a table containing all the data (a flat file)
- The database is then *defragmented*
  - The process of *normalisation*
  - There are 5 levels of normalisation
  - 3 levels of normalisation are generally found the be sufficient:
    - *First* normal form / *Second* normal form / *Third* normal form

Normalisation

- Normalisation
  - A correctly designed database is created from the ER model
  - The initial data is generally set in a single table (all the data)
  - The database is then *defragmented* (the process of *normalisation* – there are 5 levels of normalisation (of which 3 levels are considered sufficient in most cases)
    - First normal form / second normal form / third normal form
- The process of normalization is designed to remove duplication of data records in a database
  - Record duplication (of records and data) causes issues in maintaining and updating a database with potential errors

Structured Query Language (SQL)
and
SQL Statements

The Structured Query Language

- *SQL* is an abbreviation of *structured query language* and is pronounced either *see-kwell* (or) as separate letters *SQL*
  - The original version called *SEQUEL* (structured English query language) was designed by an IBM research center in the mid 1970's
- *SQL* is a standardised query language for
  - Accessing a *RDMS* (and)
  - Requesting / adding / deleting / and requesting (querying / searching) for information from a database

SQL to Create a Database Table

- The SQL shows the code to create a customer table
- The MySQL types shown are
  - char
  - varchar
  - Int
  - For other types see the course resources
- The (50) shows the length of the string allowed (50 characters)
- The int(4) specifies an int with 4 digits
- The PRIMARY KEY(cust_id) specifies the primary key for the customer table

```
CREATE TABLE customer (
cust_id int(5) NOT NULL,
surname varchar(50),
firstname varchar(50),
initial char(1),
title_id int(3),
address varchar(50),
city varchar(50),
state varchar(20),
zipcode varchar(10),
country_id int(4),
phone varchar(15),
birth_date char(10),
PRIMARY KEY (cust_id)
) ;
```

SQL Statement Syntax

- SQL statements:
  - INSERT INTO customer VALUES (1, 'Williams', 'Lucy', 'E', '2002-07-02');
  - INSERT INTO customer VALUES (2, 'JOnes', 'Thomas', 'R', '1993-05-23');
  - INSERT INTO customer VALUES (3, 'Thomas', 'Philip', 'M', '1997-11-17');
  - SELECT * FROM customer LIMIT 100;
  - SELECT * FROM customer WHERE surname='Thomas'
  - UPDATE customer SET surname = 'Jones' WHERE cust_id = 2;
- These SQL statement demonstrate the INSERT / SELECT / UPDATE statements
  - The SELECT * FROM customer selects all the records in the table (* is a 'wildcard')
- The other SQL statements follow this pattern
- Examples of SQL statements may be found in the course resources

SQL Statement Syntax

- SQL update statement:
  - UPDATE customer SET surname = 'Jones' WHERE cust_id = 2;
- The SQL statement corrects the error:
  - The text “JOnes” is corrected to “Jones”
- The SQL UPDATE SQL statement is shown in the following slides with the output in the NetBeans IDE
- We will demonstrate the use of SQL statements with worked examples

The UPDATE SQL Statement

- To demonstrate the SQL syntax and working with a MySQL database a simple single table test database was created
- The test database has a contacts table with 5 rows (or records)
- Each record has 4 attributes:
  - id (int(5))
  - f_name (varchar (50))
  - s_name (varchar(50))
  - email (varchar(50))
- The initial table was populated with a (deliberate) error (id = 1) for all records
- The following slides show the UPDATE SQL statement and the result
  - UPDATE customers SET id = '2' WHERE f_name = 'Tom'

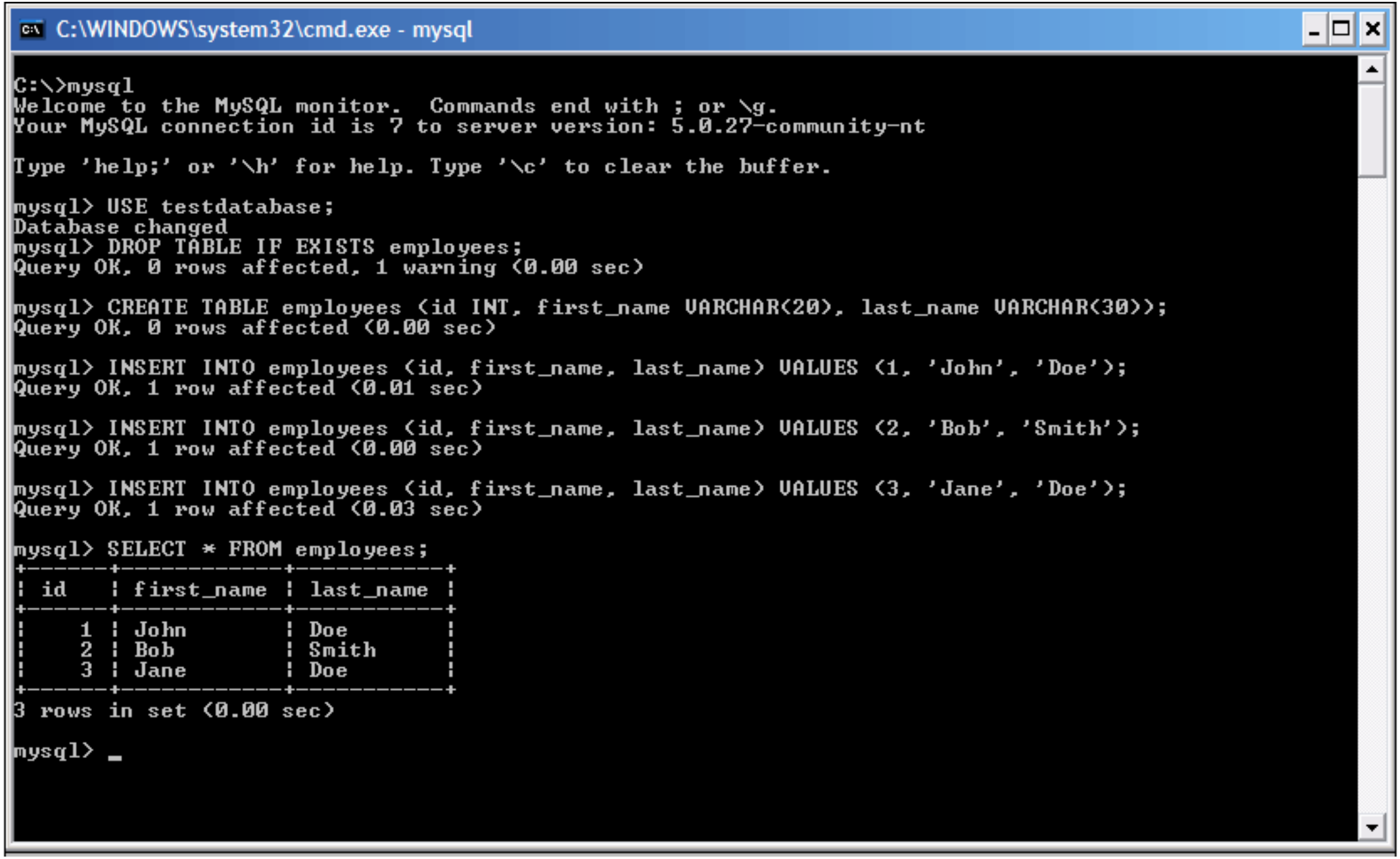
Entering SQL Statements

- Do not copy and paste the SQL statements into the editor
- Copying and pasting may introduce unwanted formatting characters
  - The SQL statement may not work (*and you will wonder why?)*
- All the SQL statements (in both MySQL and PHP):
  - Must be typed directly into the editor
  - The syntax must be perfect
  - Any deviation from the syntax will result in failure
- Remember:
  - The database must be created
  - The MySQL server must be connected
  - The XAMPP server must be running

MySQL Server using the Command line

MySQL

- The following slide shows the creation of a database table in MySQL server
- The table is populated with records (data)
- In this example the table is created using the command line
- We will use the NetBeans IDE and the XAMPP server (which includes the MySQL server)
- Shown is the SQL statements and the resulting table stored in the MySQL server


C:\WINDOWS\system32\cmd.exe - mysql
C:\>mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 5.0.27-community-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> USE testdatabase;
Database changed
mysql> DROP TABLE IF EXISTS employees;
Query OK, 0 rows affected, 1 warning (0.00 sec)
mysql> CREATE TABLE employees (id INT, first_name VARCHAR(20), last_name VARCHAR(30));
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO employees (id, first_name, last_name) VALUES (1, 'John', 'Doe');
Query OK, 1 row affected (0.01 sec)
mysql> INSERT INTO employees (id, first_name, last_name) VALUES (2, 'Bob', 'Smith');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO employees (id, first_name, last_name) VALUES (3, 'Jane', 'Doe');
Query OK, 1 row affected (0.03 sec)
mysql> SELECT * FROM employees;
+------+------------+-----------+
| id | first_name | last_name |
+------+------------+-----------+
1	John	Doe
2	Bob	Smith
3	Jane	Doe
+------+------------+-----------+
3 rows in set (0.00 sec)
mysql> _

Some Advanced MySQL Server Examples

Advanced MySQL

- We have seen the basic (and most often used) functions and techniques to access a MySQL database including user defined variables
- MySQL provides advanced SELECT and other functions to work with to mange the database tables and data
- In RBMS the results of a query is termed a 'view'
  - In a SELECT query we are generating a view of selected data
- A detailed list of MySQL functions and operators can be found in:
  - The course resources book (Sams Teach Yourself PHP, MySQL & JavaScript All in One SIXTH EDITION)
  - We have provided two MySQL tutorial documents in the course resources
  - The tutorials cover every aspect of MySQL including the functions and operators

MySQL Functions

- MySQL provides advanced functions to work with – for example:
  - Strings / dates and times / Arithmetic and comparison operators / mathematical functions / string concatenation CONCAT() / Numeric data types (signed and unsigned) / Advanced join types (e.g., SELECT from 2 or more tables in a database)
  - Using aliases
  - Nested queries
  - Other advanced clauses including:
    - IN / EXISTS / ROLLUP /GROUP BY / HAVING / IGNORE / REPLACE / LIKE / DELETE / TRIM / etc
  - Automatic (scheduled) querying
  - MySQL server and MySQL database management security and access
  - Database and MySQL server reports

MySQL DateTime Example

- MySQL DATETIME
  - DATETIME: is a date and time combination in the default format:
    - YYYY-MM-DD HH:MM:SS
    - The available range is between 1000-01-01 00:00:00 and 9999-12-31 23:59:59
    - For example: 3:30 in the afternoon of December 30, 1973 is stored as
      - **1973-12-30 15:30:00**
    - An example of the MySQL syntax:

```
mysql> SELECT something FROM tbl_name -> WHERE
DATE_SUB(CURDATE(),INTERVAL 30 DAY) <= date_col;
```

- The query also selects rows with dates / times that are current, lie in the past, or lie in the future

MySQL DateTime Example

- An example of the MySQL syntax:

```
mysql> SELECT something FROM tbl_name ->
WHERE DATE_SUB(CURDATE(),INTERVAL 30 DAY) <=
date_col;
```

MySQL Syntax Example

- Examples of the MySQL syntax:

```
mysql> SELECT something FROM tbl_name -> WHERE
DATE_SUB(CURDATE(),INTERVAL 30 DAY) <= date_col;

mysql> SELECT DAYOFMONTH('2001-11-00'), MONTH('2005-00-00'); -
> 0, 0
mysql> SELECT DATE_ADD('2006-05-00',INTERVAL 1 DAY); -> NULL
mysql> SELECT DAYNAME('2006-05-00'); -> NULL

mysql> SELECT DATE_ADD('2008-01-02', INTERVAL 31 DAY); ->
'2008-02-02' mysql> SELECT ADDDATE('2008-01-02', INTERVAL 31
DAY); -> '2008-02-02'

mysql> SELECT CURDATE(); -> '2008-06-13' mysql> SELECT
CURDATE() + 0; -> 20080613
```

Resources

- A detailed list of MySQL functions and operators can be found in:
  - The course resources book (Sams Teach Yourself PHP, MySQL & JavaScript All in One SIXTH EDITION)
- We have provided two MySQL tutorial documents in the course web site
  - The document cover most aspects of MySQL including the functions and operators

MySQL Administration Tool
and
Creating a User Account

Overview

- In this part of the tutorial we:
  - Introduce and demonstrate the MySQL administration tool
  - Show how to create a user account with a username and password
  - Introduce MySQL privileges
  - Consider the security aspects of MySQL accounts and the related privileges
  - Consider the localhost and ports as they relate to PHP and MySQL
- In the final part of this tutorial:
  - We demonstrate how to create a MySQL database
  - access and manage data in a MySQL database using a PHP script
  - We show that accessing, viewing, and updating the MySQL database is a simple operation
  - Finally, we consider how to improve the PHP script

Practical Real-World Applications

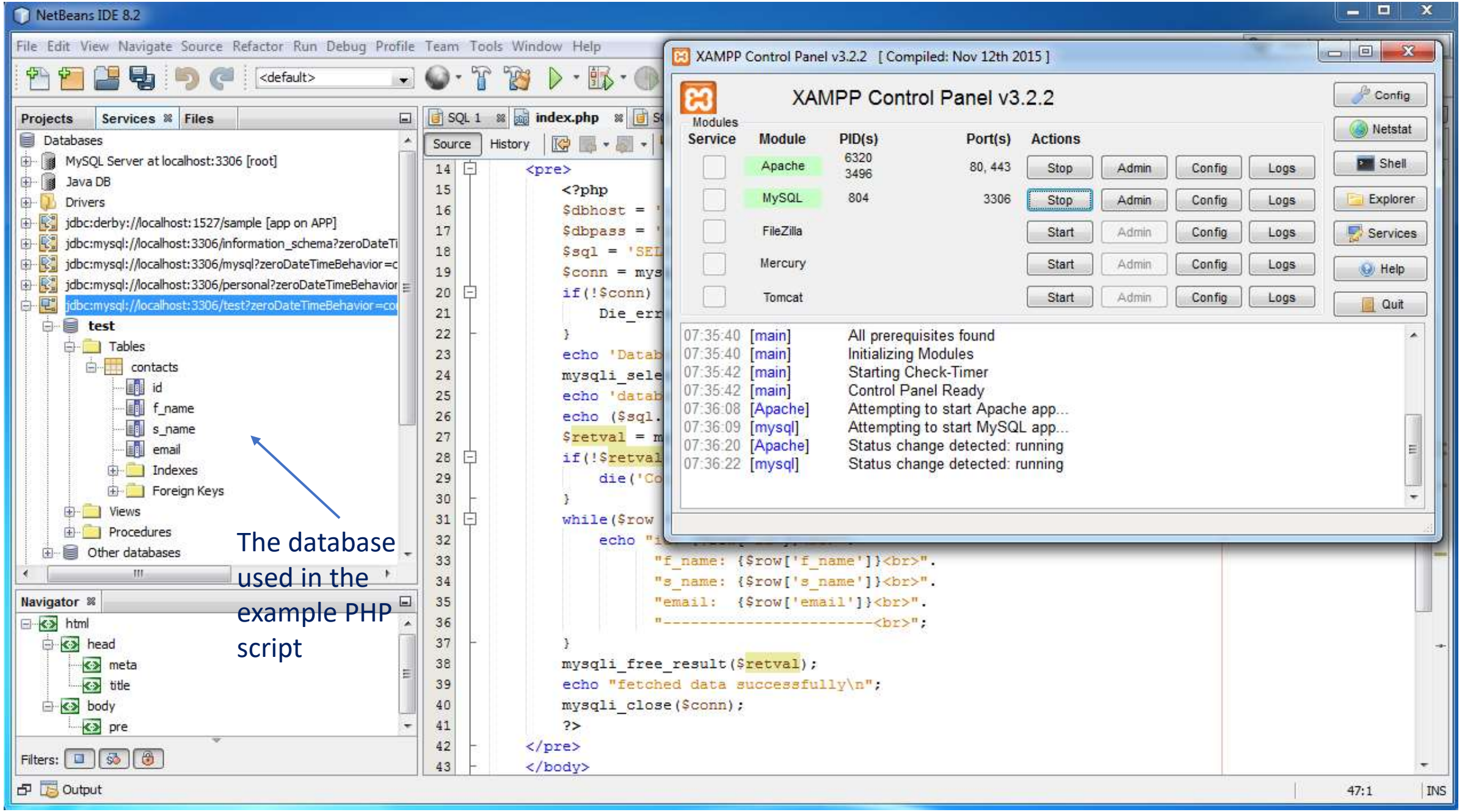
- Executing SQL in MySQL server is interesting and there are cases where the data output within the MySQL server is all that is required by an organization (no Internet access needed)
- However:
  - We are working within web-systems using web services
  - We need a way to present the results of SQL actions and queries in web-based systems
  - To achieve this, we must integrate SQL queries into web programs
  - To do this we integrate SQL statements embedded in PHP scripts

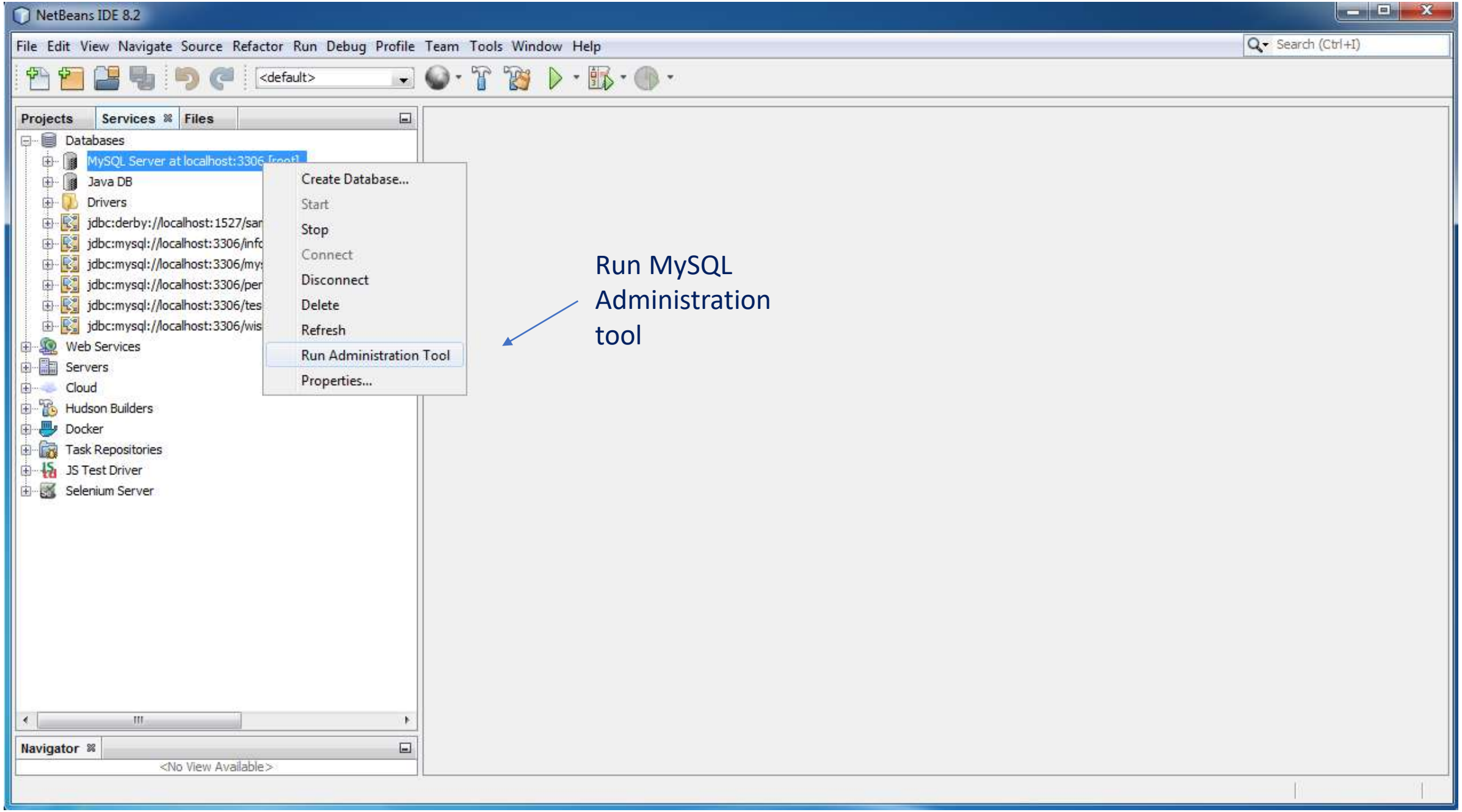
MySQL User Accounts

- We will show have how a database is created in the MySQL server using NetBeans IDE services functionality:
  - This has been achieved using the root account: MySQL Server at localhost:3306 [root]
- You must create your own personal user account with the following parameters:
  - username: (a literal string e.g., 'philip')
  - password: (a literal string – in 'real-world' systems the password will be alpha-numeric – e.g., sur7amw9cz45)
- Additionally: for the database exercises define the global privileges
- The following slides show the MySQL interface used to create use accounts and manage global privileges

MySQL Server Administration Tool

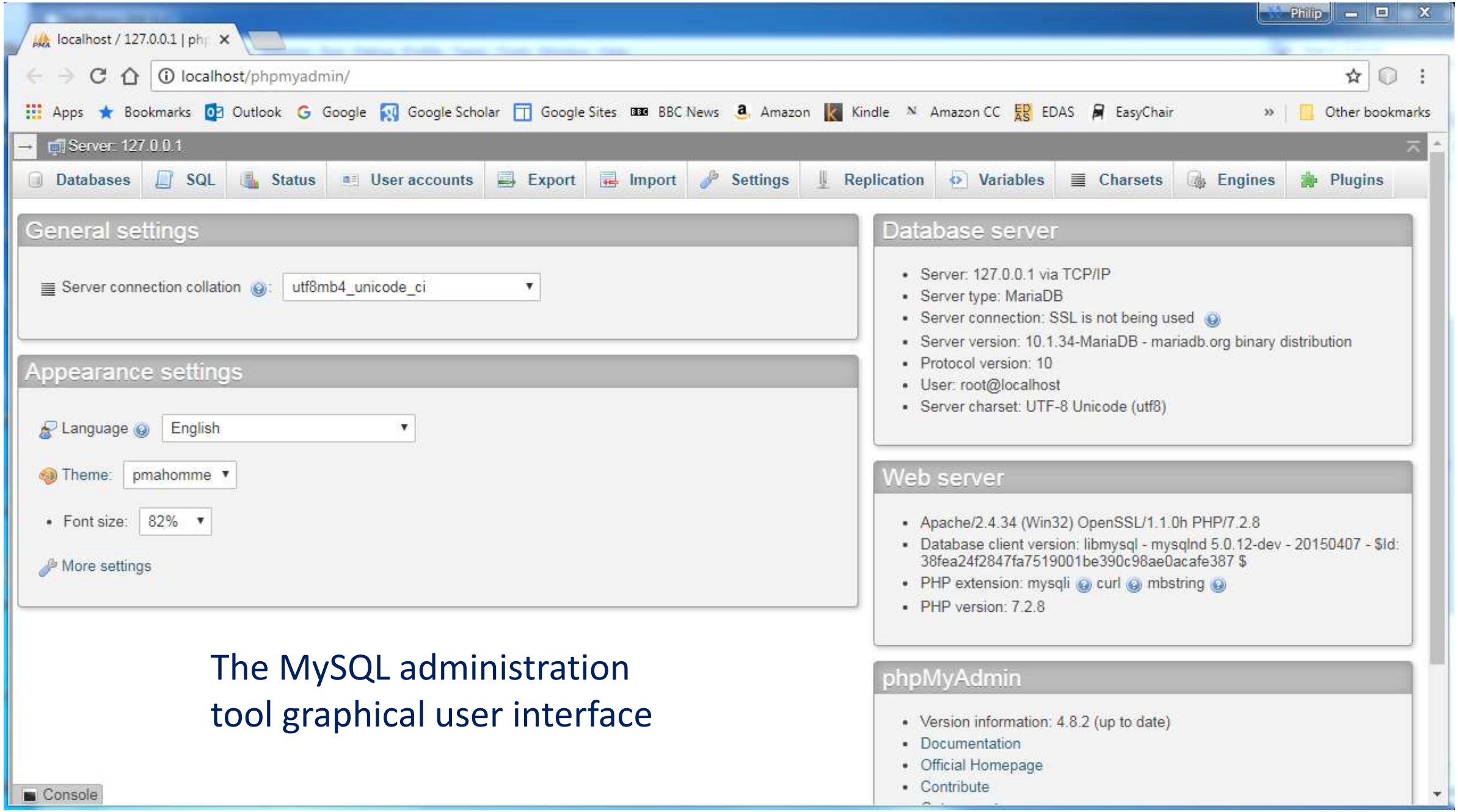
- We will demonstrate with worked examples how we can work with MySQL server in the NetBeans IDE
- We may also access the MySQL server using the MySQL administration tool
- To access the MySQL administrator tool we will use the NetBeans IDE
  - Scroll down the drop-down list and select: “Run Administration Tool”
- The following overview of the administration tool introduces:
  - My PHP script to access and operate on the MySQL database
  - The database used in my example PHP script is the test database shown in earlier slides


NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Services Files
Databases
MySQL Server at localhost:3306 [root]
Java DB
Drivers
jdbc:derby://localhost:1527/sample [app on APP]
jdbc:mysql://localhost:3306/information_schema?zeroDateTi
jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=c
jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior
jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=co
test
Tables
contacts
id
f_name
s_name
email
Indexes
Foreign Keys
Views
Procedures
Other databases
Navigator
html
head
meta
title
body
pre
Filters:
SQL 1
index.php
Source
History
14 <pre>
15 <?php
16 $dbhost = '
17 $dbpass = '
18 $sql = 'SEL
19 $conn = mys
20 if(!$conn)
21 Die_err
22 }
23 echo 'Datab
24 mysqli_sele
25 echo 'datab
26 echo ($sql.
27 $retval = m
28 if(!$retval
29 die('Co
30 }
31 while($row
32 echo "i
33 "f_name: {$row['f_name']}
".
34 "s_name: {$row['s_name']}
".
35 "email: {$row['email']}
".
36 "----------
";
37 }
38 mysqli_free_result($retval);
39 echo "fetched data successfully\n";
40 mysqli_close($conn);
41 ?>
42 </pre>
43 </body>
Output
47:1
INS
XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Modules
Service Module PID(s) Port(s) Actions
Apache 6320 3496 80, 443 Stop Admin Config Logs
MySQL 804 3306 Stop Admin Config Logs
FileZilla Start Admin Config Logs
Mercury Start Admin Config Logs
Tomcat Start Admin Config Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit
07:35:40 [main] All prerequisites found
07:35:40 [main] Initializing Modules
07:35:42 [main] Starting Check-Timer
07:35:42 [main] Control Panel Ready
07:36:08 [Apache] Attempting to start Apache app...
07:36:09 [mysql] Attempting to start MySQL app...
07:36:20 [Apache] Status change detected: running
07:36:22 [mysql] Status change detected: running
The database used in the example PHP script

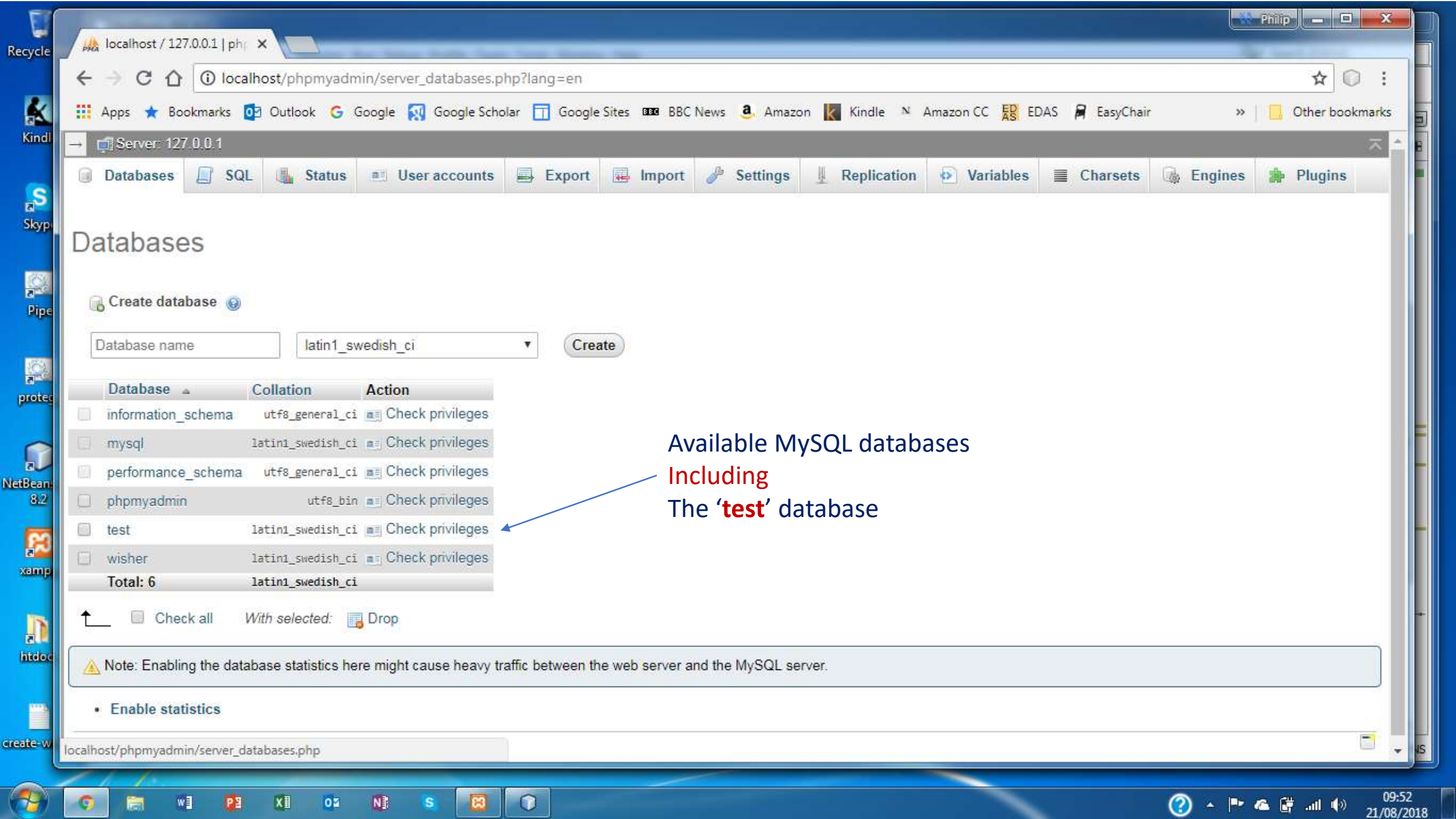

NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default>
Projects
Services
Files
Databases
MySQL Server at localhost:3306 [root]
Java DB
Drivers
jdbc:derby://localhost:1527/sar
jdbc:mysql://localhost:3306/info
jdbc:mysql://localhost:3306/my
jdbc:mysql://localhost:3306/per
jdbc:mysql://localhost:3306/tes
jdbc:mysql://localhost:3306/wis
Web Services
Servers
Cloud
Hudson Builders
Docker
Task Repositories
JS Test Driver
Selenium Server
Create Database...
Start
Stop
Connect
Disconnect
Delete
Refresh
Run Administration Tool
Properties...
Navigator
<No View Available>
Run MySQL Administration tool

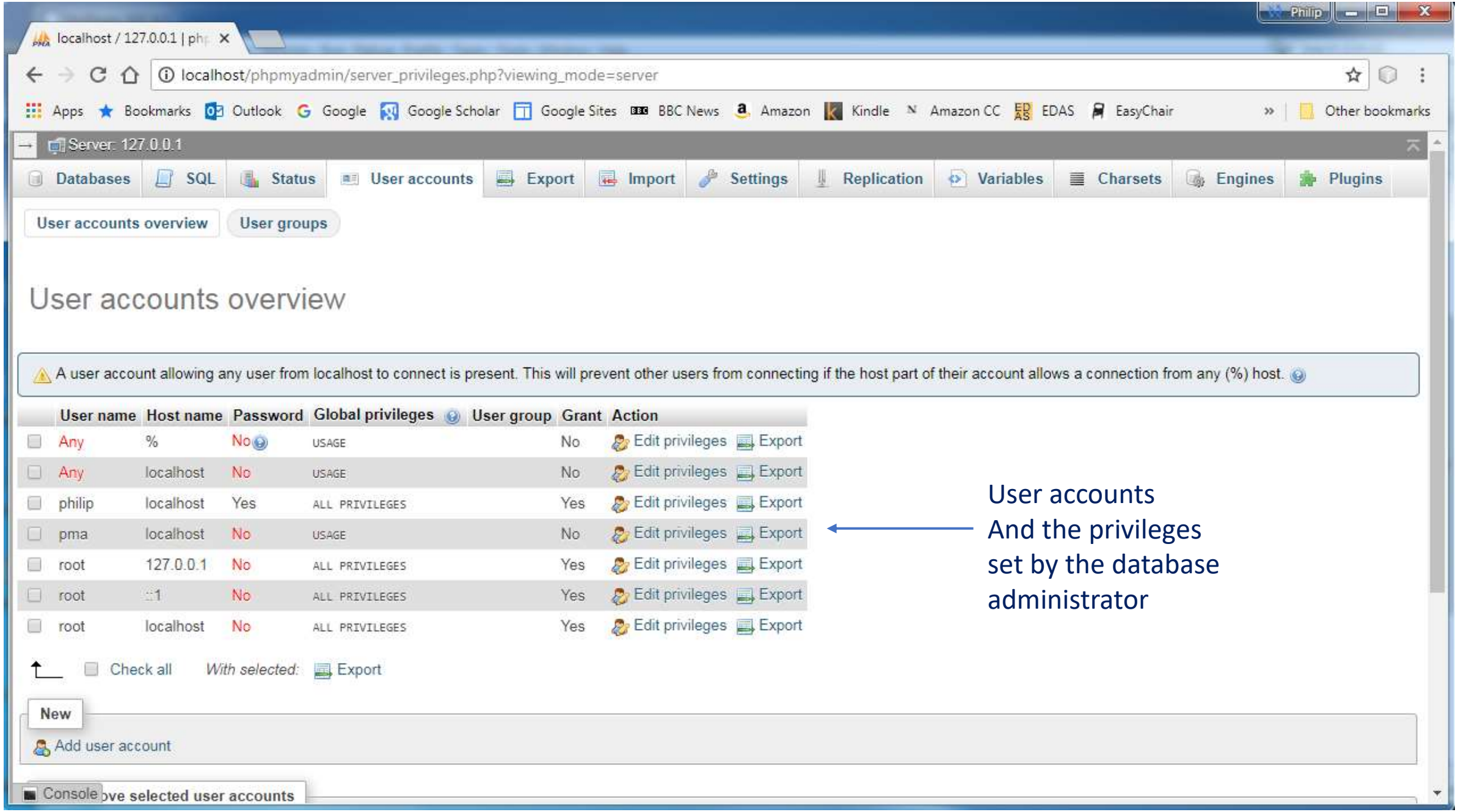
MySQL Server Administration Tool

- Running the administration tool we see the initial graphical user interface
- There are a number of tabs to task-specific interfaces where we can:
  - Manage the MySQL server
  - Create and manage databases
  - Run SQL queries
  - Create and manage user accounts with privileges



The MySQL administration tool graphical user interface

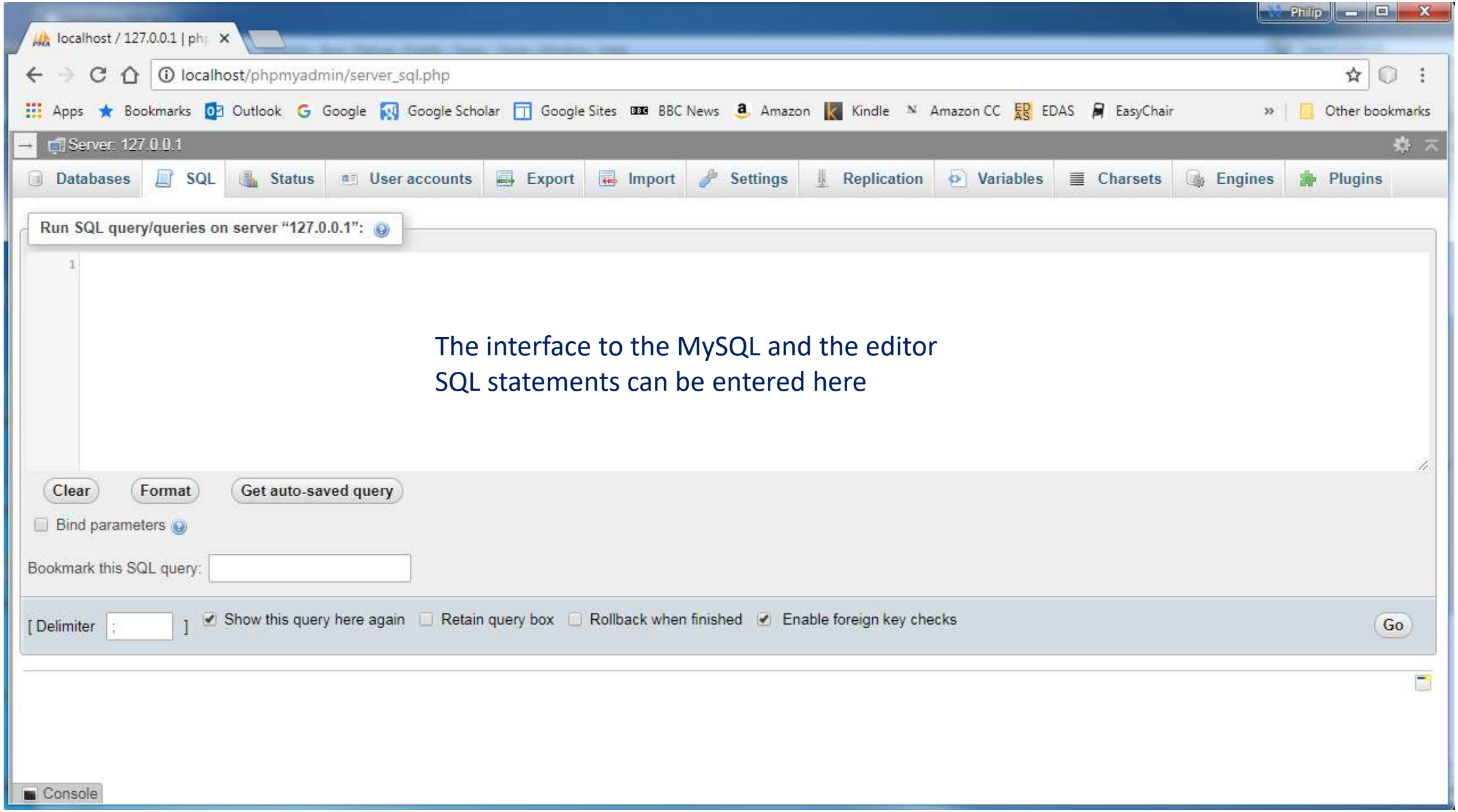

localhost / 127.0.0.1 | ph
localhost/phpmyadmin/server_databases.php?lang=en
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
Databases
Create database
Database name
latin1_swedish_ci
Create
Database
Collation
Action
information_schema utf8_general_ci Check privileges
mysql latin1_swedish_ci Check privileges
performance_schema utf8_general_ci Check privileges
phpmyadmin utf8_bin Check privileges
test latin1_swedish_ci Check privileges
wisher latin1_swedish_ci Check privileges
Total: 6 latin1_swedish_ci
Check all
With selected:
Drop
Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.
Enable statistics
Available MySQL databases
Including
The 'test' database



User accounts
And the privileges
set by the database
administrator

localhost:3306 [root]

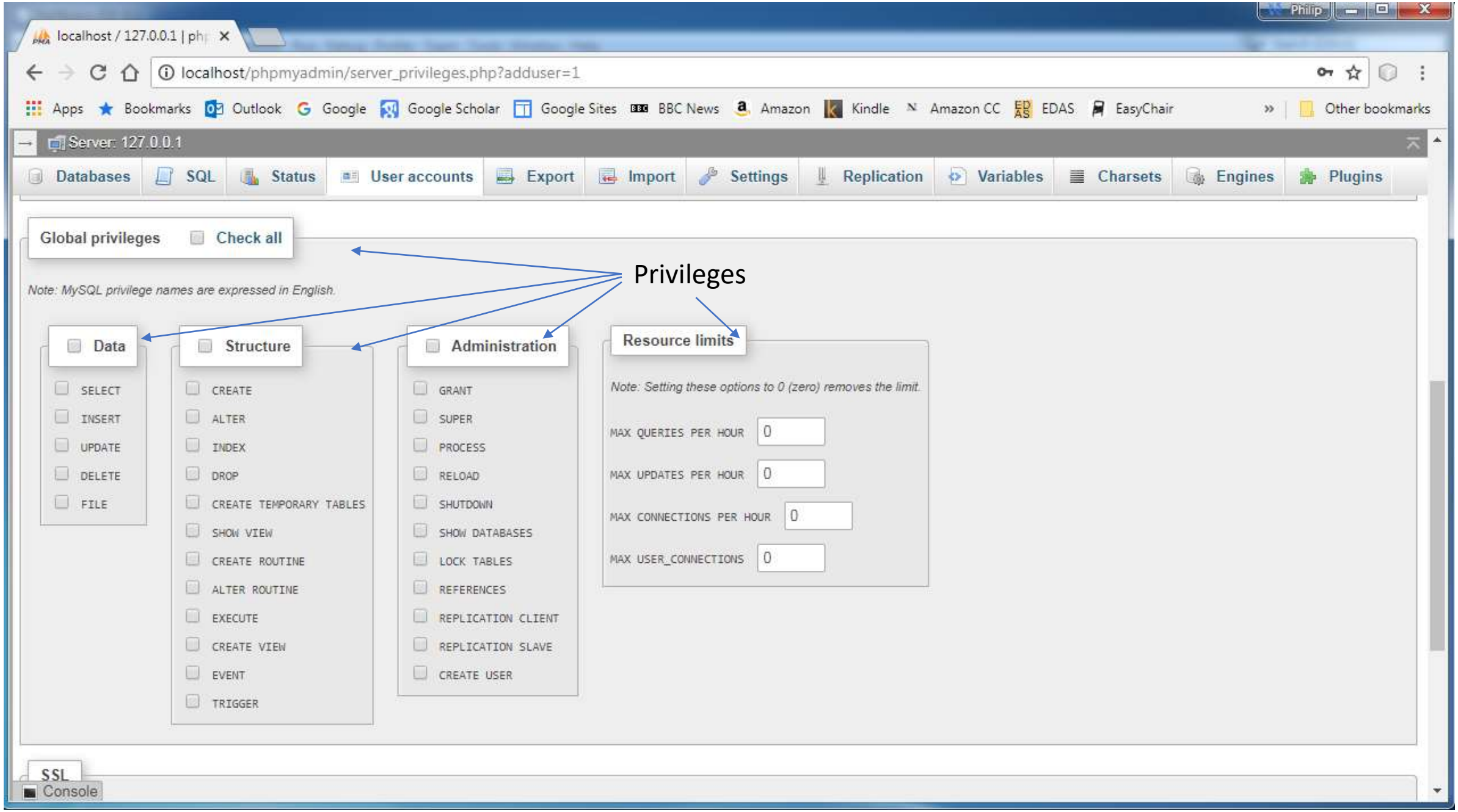
- The database is created using two parameters:
  - [root] (the 'owner' of the database)
  - localhost:3306 (the port number used)
- In a 'real-world' MySQL database:
  - The parameter [root] would not be used (it is generally deleted from MySQL server by the database admin)
  - The port would be changed from the default (3306) to another port defined in the MySQL server
  - These changes are made by the MySQL administrator for security reasons

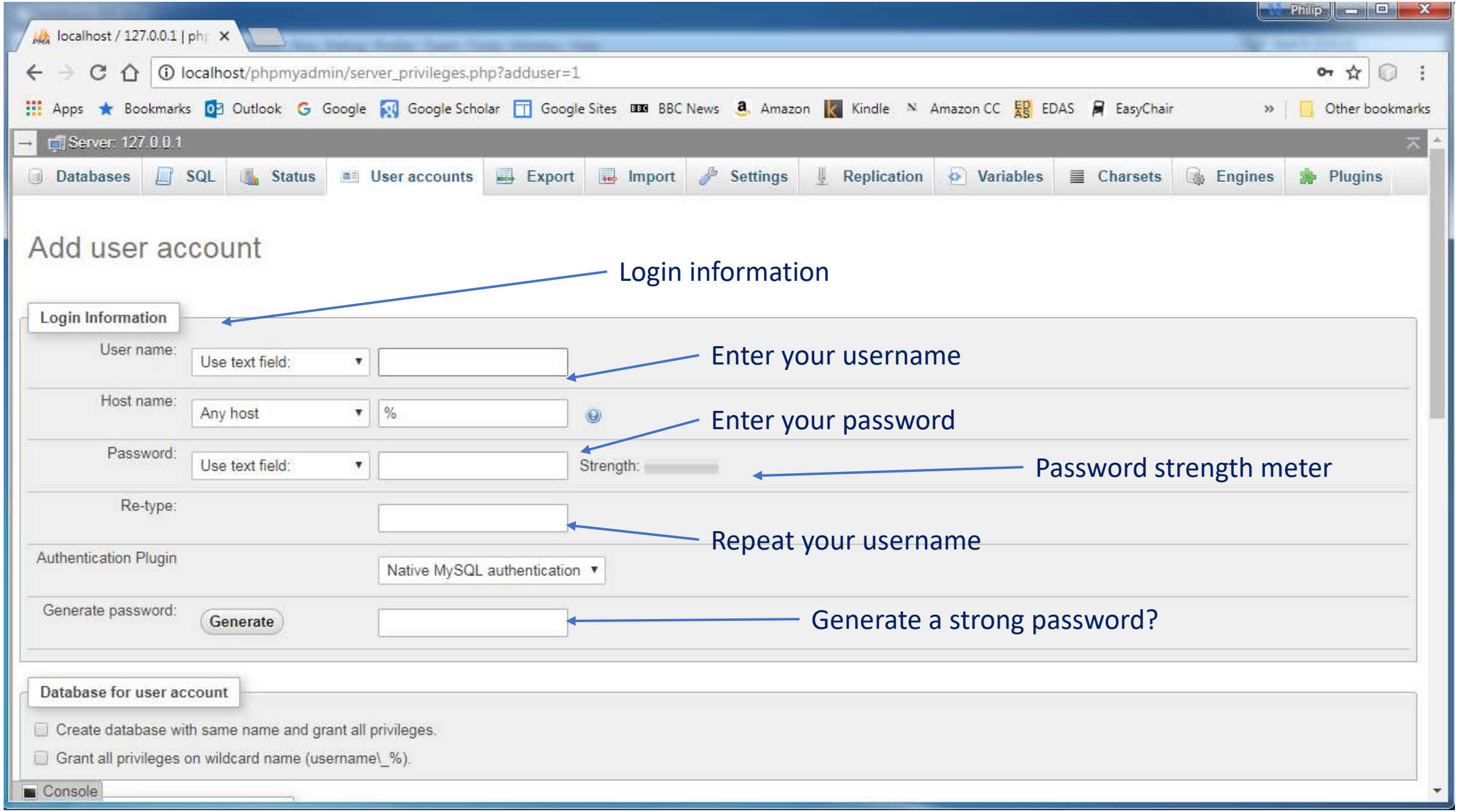


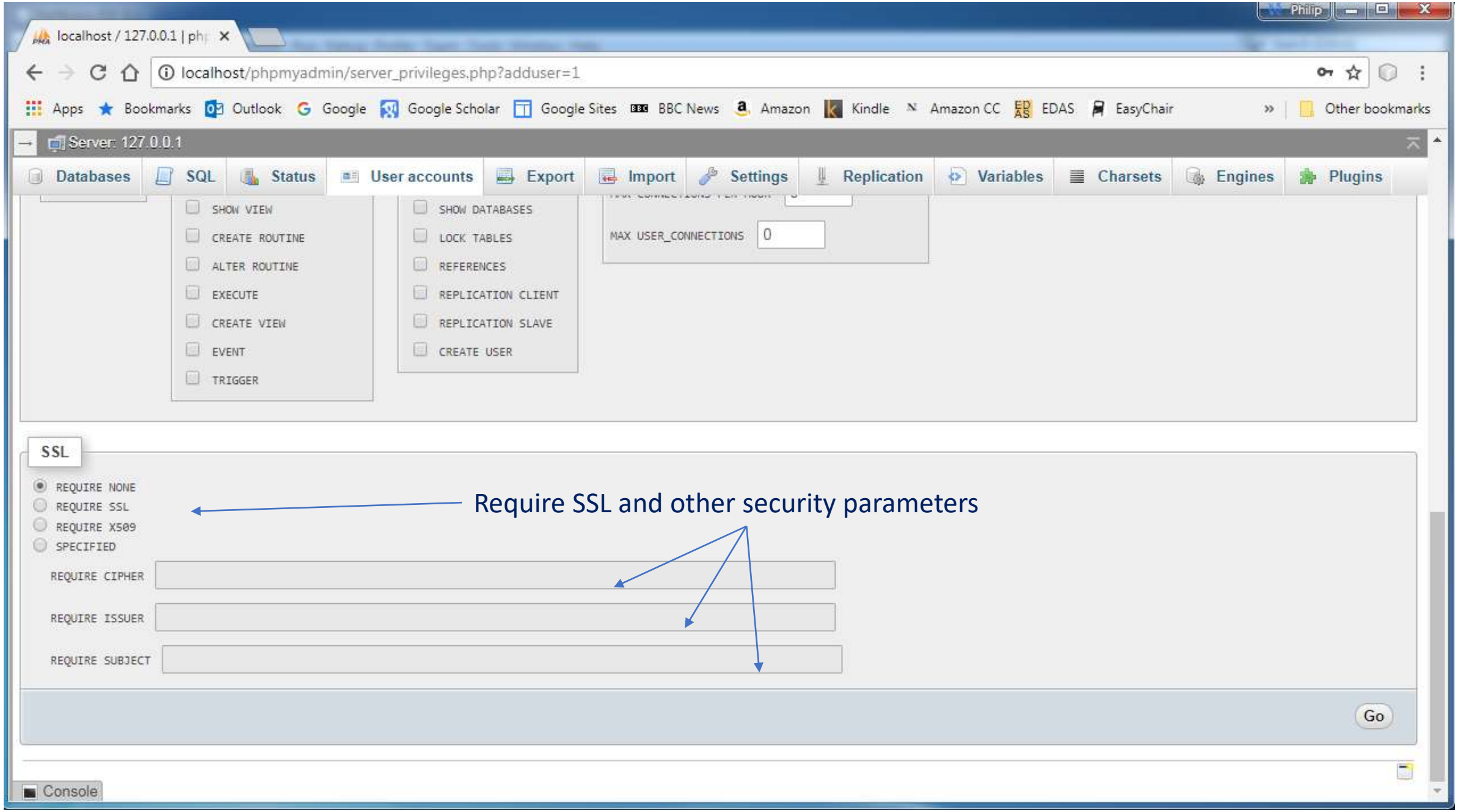
The interface to the MySQL and the editor
SQL statements can be entered here

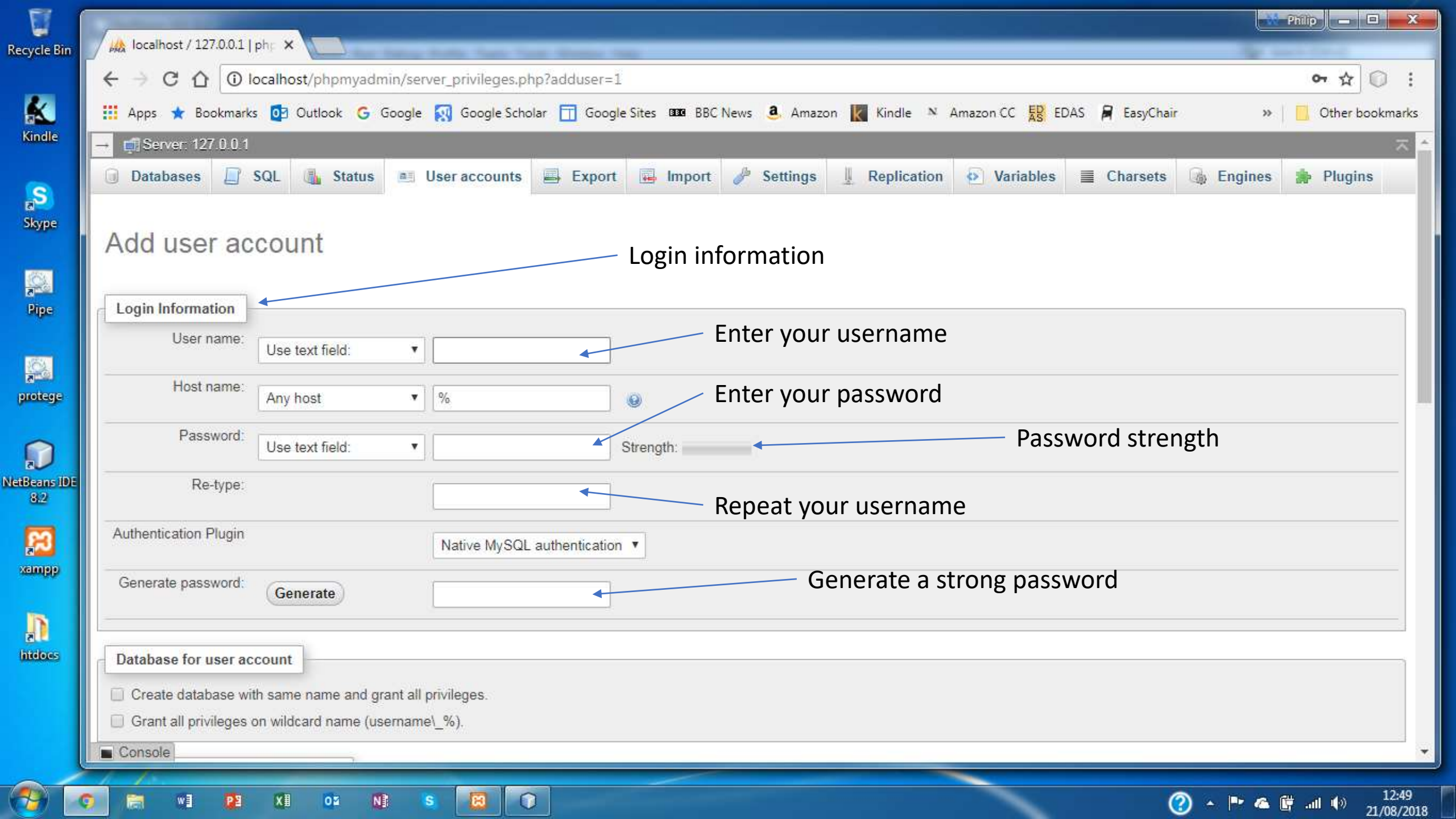
MySQL User Accounts

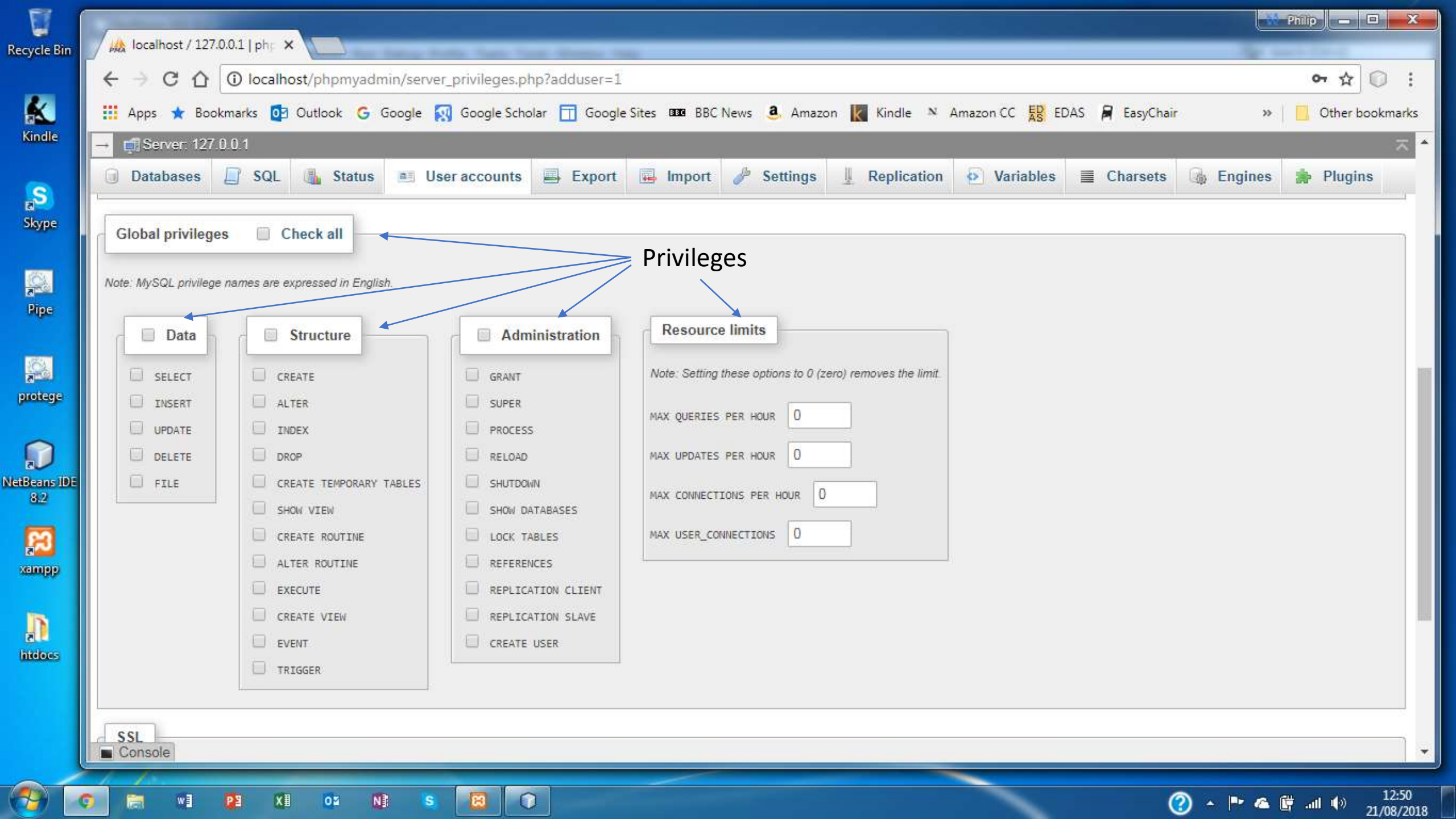
- We have seen how a database is created in the MySQL server using NetBeans IDE services functionality
  - This has been achieved using the root account: MySQL Server at localhost:3306 [root]
- Now: you must create your personal user account with the following parameters:
  - **username**: (a literal string e.g., 'philip')
  - **password**: (a literal string – in 'real-world' systems the password will be alpha-numeric – such as: sur7amw9cz45)
  - Additionally: you must define global privileges for this exercise
- The following slides show the MySQL interface used to create use accounts and manage global privileges

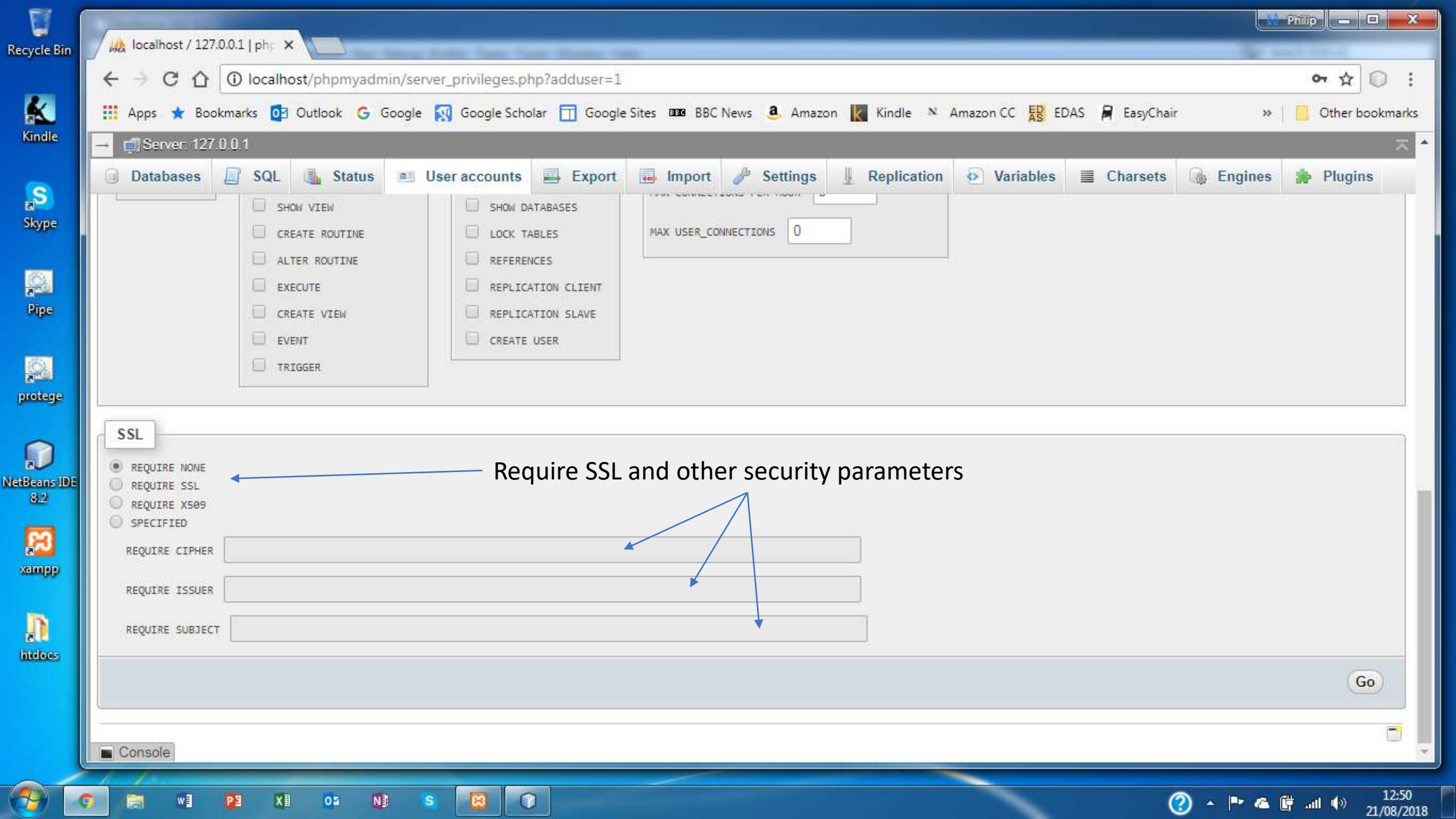

localhost / 127.0.0.1 | ph
localhost/phpmyadmin/server_privileges.php?adduser=1
Apps
Bookmarks
Outlook
Google
Google Scholar
Google Sites
BBC News
Amazon
Kindle
Amazon CC
EDAS
EasyChair
Other bookmarks
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
Global privileges
Check all
Note: MySQL privilege names are expressed in English.
Privileges
Data
SELECT
INSERT
UPDATE
DELETE
FILE
Structure
CREATE
ALTER
INDEX
DROP
CREATE TEMPORARY TABLES
SHOW VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE
CREATE VIEW
EVENT
TRIGGER
Administration
GRANT
SUPER
PROCESS
RELOAD
SHUTDOWN
SHOW DATABASES
LOCK TABLES
REFERENCES
REPLICATION CLIENT
REPLICATION SLAVE
CREATE USER
Resource limits
Note: Setting these options to 0 (zero) removes the limit.
MAX QUERIES PER HOUR 0
MAX UPDATES PER HOUR 0
MAX CONNECTIONS PER HOUR 0
MAX USER_CONNECTIONS 0
SSL
Console


localhost / 127.0.0.1 | ph
localhost/phpmyadmin/server_privileges.php?adduser=1
Apps
Bookmarks
Outlook
Google
Google Scholar
Google Sites
BBC News
Amazon
Kindle
Amazon CC
EDAS
EasyChair
Other bookmarks
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
Add user account
Login information
Login Information
User name:
Use text field:
Enter your username
Host name:
Any host
%
Password:
Use text field:
Enter your password
Strength:
Password strength meter
Re-type:
Repeat your username
Authentication Plugin
Native MySQL authentication
Generate password:
Generate
Generate a strong password?
Database for user account
Create database with same name and grant all privileges.
Grant all privileges on wildcard name (username_%).
Console


localhost / 127.0.0.1 | ph
localhost/phpmyadmin/server_privileges.php?adduser=1
Apps
Bookmarks
Outlook
Google
Google Scholar
Google Sites
BBC News
Amazon
Kindle
Amazon CC
EDAS
EasyChair
Other bookmarks
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
SHOW VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE
CREATE VIEW
EVENT
TRIGGER
SHOW DATABASES
LOCK TABLES
REFERENCES
REPLICATION CLIENT
REPLICATION SLAVE
CREATE USER
MAX USER_CONNECTIONS
0
SSL
REQUIRE NONE
REQUIRE SSL
REQUIRE X509
SPECIFIED
REQUIRE CIPHER
REQUIRE ISSUER
REQUIRE SUBJECT
Require SSL and other security parameters
Go
Console


Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
localhost / 127.0.0.1 | php
localhost/phpmyadmin/server_privileges.php?adduser=1
Apps
Bookmarks
Outlook
Google
Google Scholar
Google Sites
BBC News
Amazon
Kindle
Amazon CC
EDAS
EasyChair
Other bookmarks
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
Add user account
Login information
Login Information
User name:
Use text field:
Enter your username
Host name:
Any host
%
Password:
Use text field:
Enter your password
Strength:
Password strength
Re-type:
Repeat your username
Authentication Plugin
Native MySQL authentication
Generate password:
Generate
Generate a strong password
Database for user account
Create database with same name and grant all privileges.
Grant all privileges on wildcard name (username_%).
Console
12:49
21/08/2018


Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
localhost / 127.0.0.1 | php
localhost/phpmyadmin/server_privileges.php?adduser=1
Apps
Bookmarks
Outlook
Google
Google Scholar
Google Sites
BBC News
Amazon
Kindle
Amazon CC
EDAS
EasyChair
Other bookmarks
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
Global privileges
Check all
Note: MySQL privilege names are expressed in English.
Privileges
Data
SELECT
INSERT
UPDATE
DELETE
FILE
Structure
CREATE
ALTER
INDEX
DROP
CREATE TEMPORARY TABLES
SHOW VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE
CREATE VIEW
EVENT
TRIGGER
Administration
GRANT
SUPER
PROCESS
RELOAD
SHUTDOWN
SHOW DATABASES
LOCK TABLES
REFERENCES
REPLICATION CLIENT
REPLICATION SLAVE
CREATE USER
Resource limits
Note: Setting these options to 0 (zero) removes the limit.
MAX QUERIES PER HOUR 0
MAX UPDATES PER HOUR 0
MAX CONNECTIONS PER HOUR 0
MAX USER_CONNECTIONS 0
SSL
Console
12:50
21/08/2018


Recycle Bin
Kindle
Skype
Pipe
protege
NetBeans IDE 8.2
xampp
htdocs
localhost / 127.0.0.1 | php
localhost/phpmyadmin/server_privileges.php?adduser=1
Apps
Bookmarks
Outlook
Google
Google Scholar
Google Sites
BBC News
Amazon
Kindle
Amazon CC
EDAS
EasyChair
Other bookmarks
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
Charsets
Engines
Plugins
SHOW VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE
CREATE VIEW
EVENT
TRIGGER
SHOW DATABASES
LOCK TABLES
REFERENCES
REPLICATION CLIENT
REPLICATION SLAVE
CREATE USER
MAX USER_CONNECTIONS
0
SSL
REQUIRE NONE
REQUIRE SSL
REQUIRE X509
SPECIFIED
Require SSL and other security parameters
REQUIRE CIPHER
REQUIRE ISSUER
REQUIRE SUBJECT
Go
Console
12:50
21/08/2018

Review

- In todays tutorial we have:
  - Reviewed web-based systems introduced in week #1
  - Considered the practical ‘real-world’ use of a database and briefly considered security for database applications
  - Introduced database basics and the Structured Query Language (SQL)
  - Shown how to use the MySQL Administration interface tool to:
    - Create the required user account
- In the next tutorial we will:
  - Show how to create a database with access and management using a PHP script

Practical Exercise

- We have demonstrated how to use the MySQL Administration interface tool to create the required user account
- Your task:
  - Using NetBeans access the MySQL Administration Tool
  - Create your personal user account within the MySQL Server
- The user account is required for the next steps in this week's tutorials and practical exercises
- Complete this task for the next tutorial