Chapter - 3

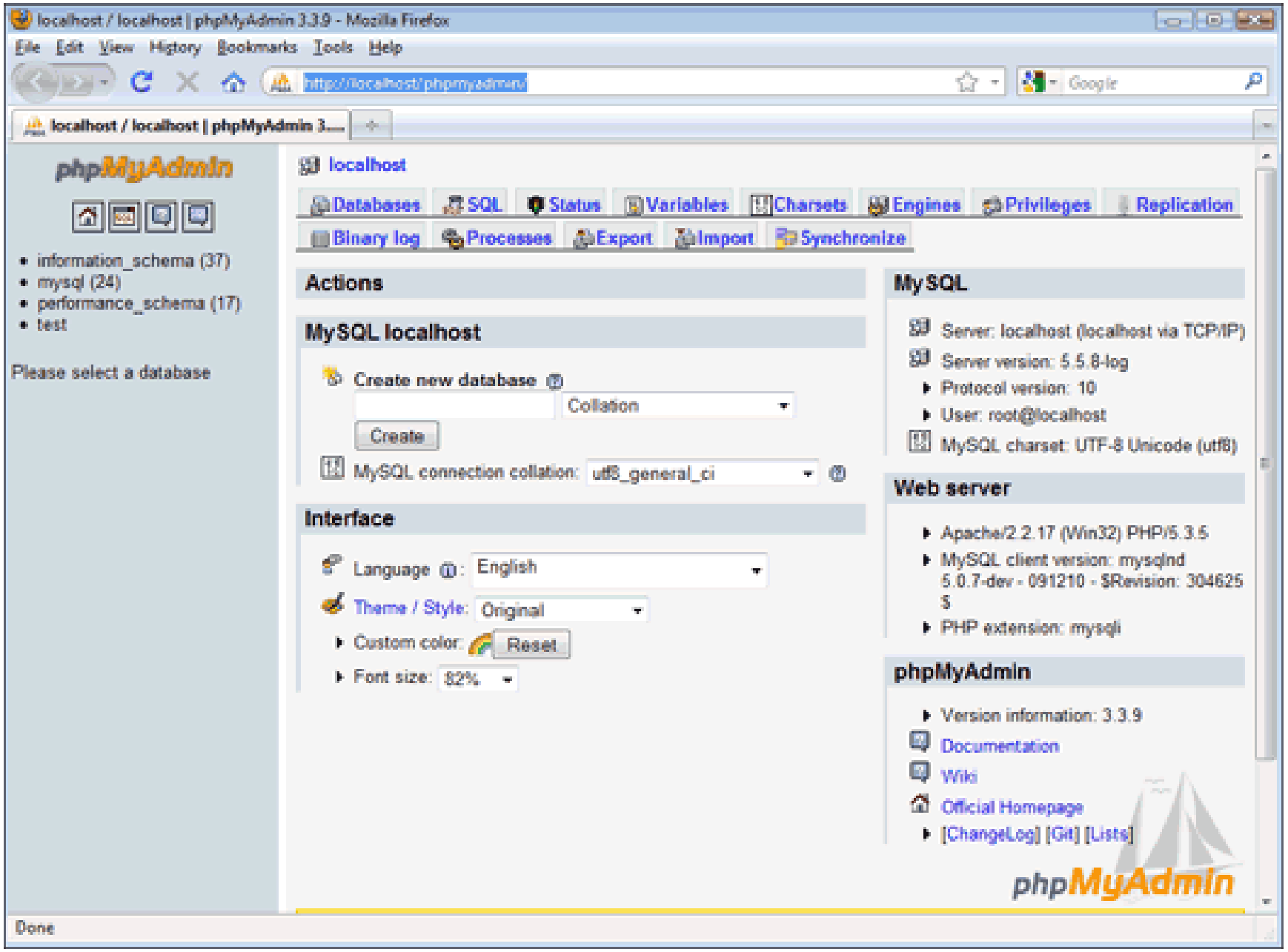
Accessing MySQL with PHP

Index

- MySQL Structure
- Basic functionalities of phpMyAdmin
- Connectivity
- Querying the database

Introduction

- The most popular database system that is used with PHP is called MySQL.
- We will create our databases with phpMyAdmin, which is part of MySQL.
- It can be reached by the URL: http://localhost:81/phpmyadmin


localhost / localhost | phpMyAdmin 3.3.9 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/phpmyadmin/
Google
localhost / localhost | phpMyAdmin 3....
phpMyAdmin
information_schema (37)
mysql (24)
performance_schema (17)
test
Please select a database
localhost
Databases SQL Status Variables Charsets Engines Privileges Replication
Binary log Processes Export Import Synchronize
Actions
MySQL localhost
Create new database
Collation
Create
MySQL connection collation: utf8_general_ci
Interface
Language : English
Theme / Style: Original
Custom color: Reset
Font size: 82%
MySQL
Server: localhost (localhost via TCP/IP)
Server version: 5.5.8-log
Protocol version: 10
User: root@localhost
MySQL charset: UTF-8 Unicode (utf8)
Web server
Apache/2.2.17 (Win32) PHP/5.3.5
MySQL client version: mysqlnd 5.0.7-dev - 091210 - $Revision: 304625 $
PHP extension: mysqli
phpMyAdmin
Version information: 3.3.9
Documentation
Wiki
Official Homepage
[ChangeLog] [Git] [Lists]
phpMyAdmin
Done

MySQL Structure

- A database is a structure that comes in two flavors: a flat database and a relational database.
- A relational database is much more oriented to the human mind and is often preferred over the flat database that are just stored on hard drives like a text file.
- MySQL is a relational database.
- MySQL is currently the most popular open source database server in existence.

Create a database with phpMyAdmin

- You can create all of your database tables and queries using PHP code.
- But before doing that, it's a good idea to get an understanding of just what it is you'll be creating.

What is a database and what do they look like?

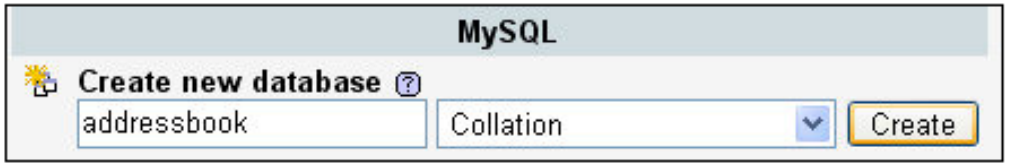
- A database is a way to store lots of information.
- You might want to store the names and addresses of all your contacts, or save usernames and passwords for your online forum.
- Or maybe customer information.
- When you create a database, you're creating a structure like this:

ID	Title	First_Name	Surname
1	Mr	Test	Name
2	Mrs	Second	Test

- The columns (ID, Title, First_Name, Surname) are called **Fields**. The rows are called **Records**. Each record is a separate entry.

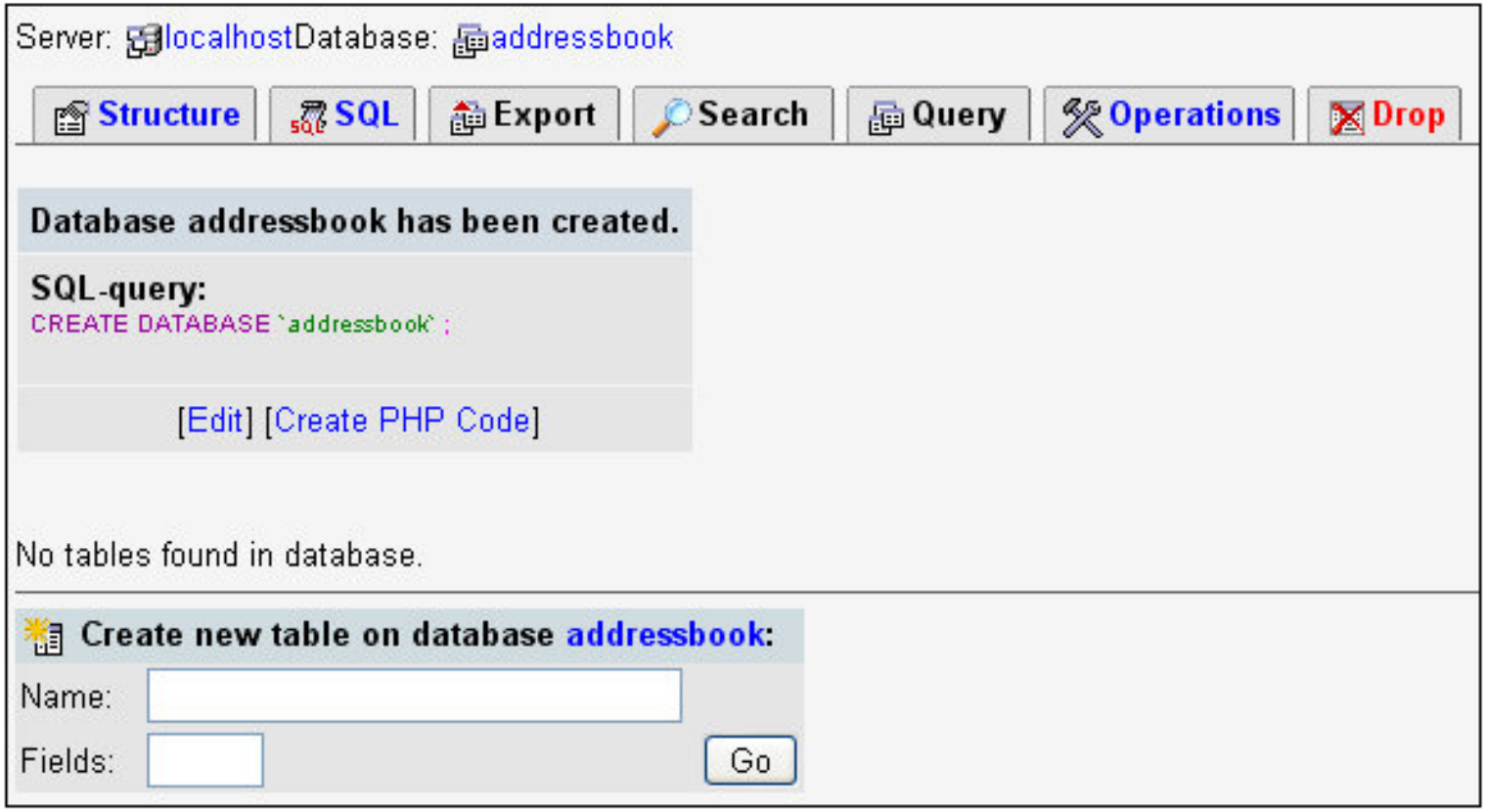
Starting with phpMyAdmin

- In a database, you save the information in a **Table**.
- A single database can contain many tables, and they can be linked together.
- When the tables are linked together, it's said to be a **relational** database.
- If you just have a single table in your database, then it's called a **flat-file** database.
- Flat-file database are easier to create and understand, so we'll start by creating one of these using phpMyAdmin.
- So bring up phpMyAdmin and click on Databases.



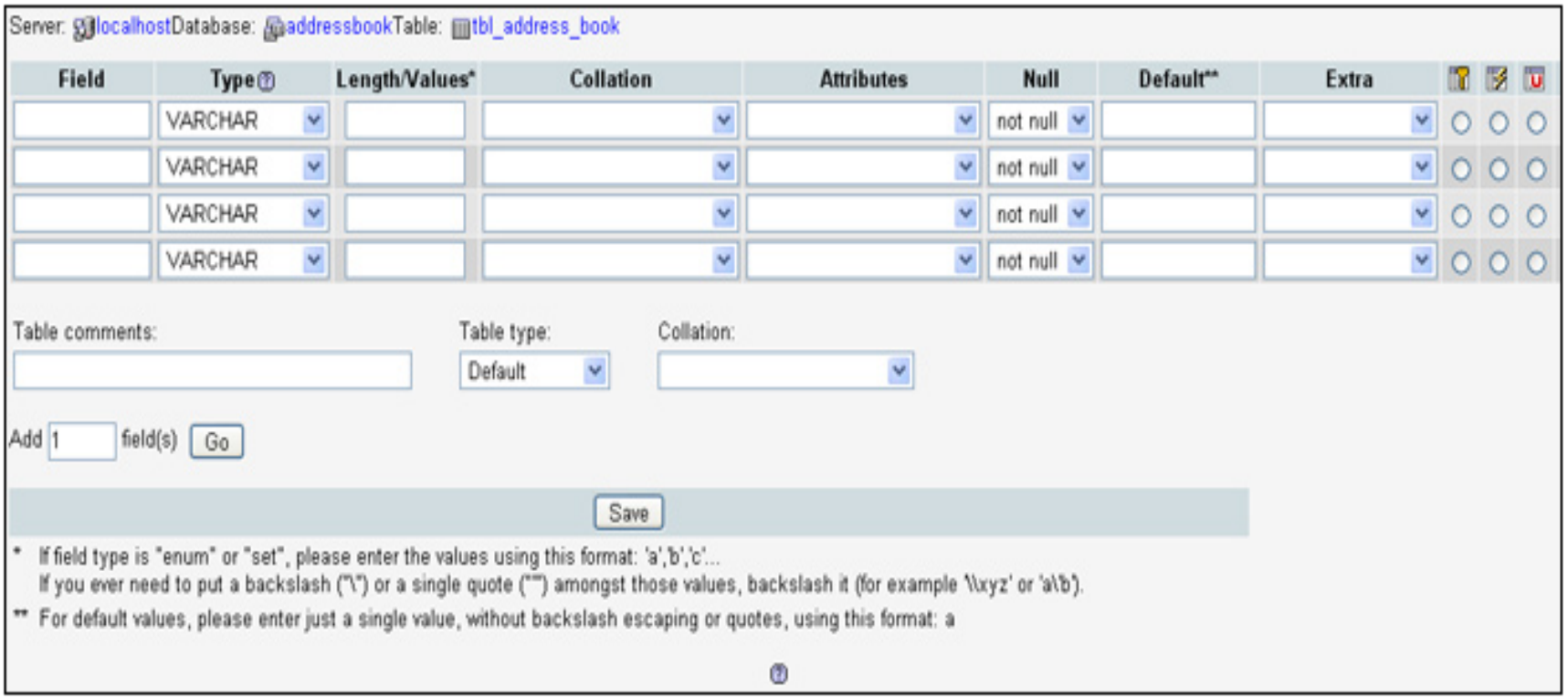
STARTING WITH PHPMYADMIN

- After you have typed a name for your new database, click the "Create" button. You will be taken to a new area.
- In this new area, you can create a Table to go in your database.



Starting with phpMyAdmin

- When you've finished, click the **Go** button. Another, more complex, area will appear.



- In this new area, you set up the **fields** in your database. You can specify whether a field is for text, for numbers, for yes/no values, etc.

MySQL Table Types

- Data in MySQL is stored in files (or memory) using a variety of different techniques.
- Each of these techniques employ different storage mechanisms, indexing facilities, locking levels and ultimately provide a range of different functions and capabilities.
- By choosing a different technique you can gain additional speed or functionality benefits that will improve the overall functionality of your application.

MySQL Table Types

- The table types are available in MySQL are:
  - --ISAM, deprecated from version 5.0
  - MyISAM
  - InnoDB
  - BerkeleyDB (BDB)
  - MERGE
  - --MEMORY , previously called HEAP i.e. deprecated
- The most important feature to make all the table types above distinctive is transaction-safe or not.
- Only InnoDB and BDB tables are transaction safe and only MyISAM tables support full-text indexing and searching feature.
- MyISAM is also the default table type when you create table without declaring which storage engine to use.

MySQL Table Types

Name	MyISAM	ISAM	InnoDB
Introduced	V3.23	V1.0	V3.23- source V4.0- +binary
Default Install	Yes	Yes	No
Data Limitation	None	Max. Size 4GB	No
Index Limitation	Max64 per table Max 16 per coln	Max 16 indexes per table	None
Transaction Support	No	No	Yes(ACID)
Locking level	Table	Table	Row

MySQL Table Types

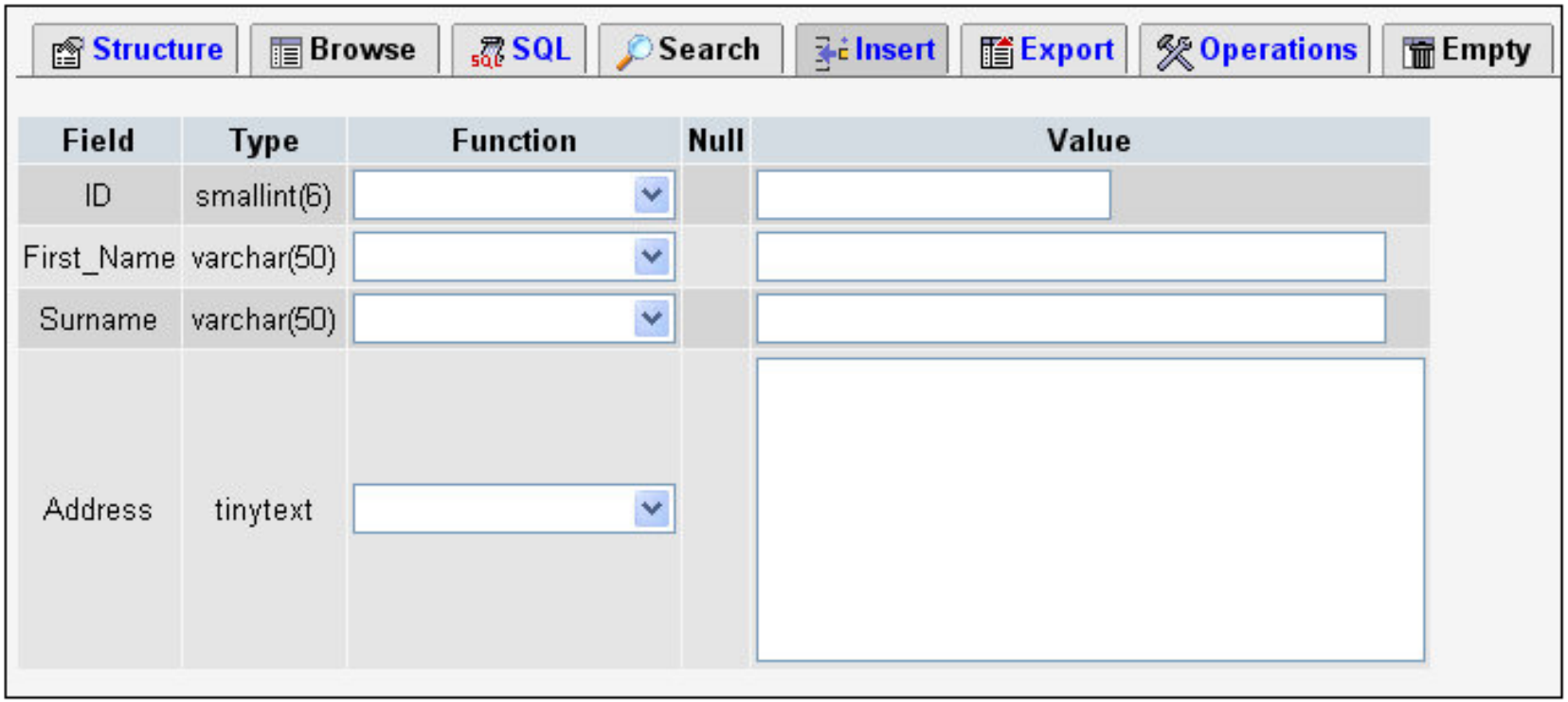
Name	BDB	Merge	Memory(Heap)
Introduced	V3.23.4a	V3.23.25	V1.0 known as Memory from 4.1
Default Install	No	Yes	Yes
Data Limitation	None	Underlying table must be MyISAM	BLOB and Text not supported
Index Limitation	Max31 per table Max 16 per coln Max key size 1024 bye	N/A	None
Transaction Support	Yes	No	No
Locking level	Page(8192 bytes)	Table	Table

INSERTING DATA

- To insert a new record to the table you created, select the **Insert** link at the top of the page:

INSERTING DATA

- Next you will se following screen, where you can inserts as many records you require.



Where to fond the tables?

- In MySQL tables are stored at C:\xampp\mysql\data – when you are using xampp server.

Opening a Connection to a MySQL database

- It's a three step process:
  1. Open a connection to MySQL itself
  2. Specify the database we want to open
  3. Close the connection

Structured Query Language

- SQL is a way to query and manipulate databases.
- If you want to grab all of the records from a table in a database, you use the **SELECT** word. Like this:
  - SELECT * FROM Table_Name;
- SQL is not case sensitive, so the above line could be written:
  - Select * From Table_Name;
- But your SQL statements are easier to read if you type the keywords in uppercase letters.
- You don't have to select all the records from your database. You can just select the columns that you need. E.g.
  - SELECT First_Name, Surname FROM tbl_address_book";

Some MySQL functions

- **mysql_connect([$host[, $username[, $password]]])** - Open a connection to a MySQL Server.
- Opens or reuses a connection to a MySQL server.
- **mysql_select_db($database[,$resource])** - Select a MySQL database.
- Sets the current active database on the server that's associated with the specified link identifier.
- Every subsequent call to **mysql_query()** will be made on the active database.
- **mysql_query($query[, $resource])** — Send a MySQL query.

Some MySQL functions

- **mysql_fetch_assoc($result)** — Fetch a result row as an associative array.
- **mysql_fetch_array($result)** — Fetch a result row as an associative array, a numeric array, or both.
  - result_type, second parameter is the type of array that is to be fetched. It's a constant and can take the following values:
    - MYSQL_ASSOC,MYSQL_NUM, and MYSQL_BOTH.
- **mysql_close()** — Close a non-persistent MySQL connection.
- **mysql_drop_db()** — Drop (delete) a MySQL database.

Some MySQL functions

- mysql_list_dbs — List databases available on a MySQL server.
- mysql_list_fields — List MySQL table fields.
- mysql_list_tables() — List tables in a MySQL database.
- mysql_num_fields() — Get number of fields in result.
- mysql_num_rows() — Get number of rows in result.
- mysql_free_result() — Free result memory.

MySQL Syntax and Commands

- Common commands
  - CREATE : Creates new databases and tables
  - ALTER : Modifies existing tables
  - SELECT : Chooses the data you want
  - DELETE : Erases the data from your table
  - DESCRIBE : Lets you know the structure and specifics of the table
  - INSERT INTO tablename VALUES : Puts values into the table
  - UPDATE : Lets you modify data already in a table
  - DROP : Deletes an entire table or database

PHP WITH MYSQL

- You can use MySQL commands within PHP code.
- Some of the more commonly used functions are:
  - mysql_connect([$host[, $username[, $password]]]) : Connects to the MySQL server and returns a resource which is used to reference the connection.
  - mysql_select_db($database[, $resource]) : Equivalent to the MySQL command USE and sets the active database.
  - mysql_query($query[, $resource]) : Used to send any MySQL command to the database server. In the case of SELECT queries, a reference to the result set will be returned.
  - mysql_fetch_array($result) : Return a row of data from the query ' s result set as an associative array, numeric array or both

PHP WITH MYSQL

- mysql_fetch_assoc($result) : Return a row of data from the query ' s result set as an associative array.
- mysql_error([$resource]) : Shows the error message generated by the previous query.

- Extra :-
- die(<String> $msg);
  - die() prints message and exits the current script.

Example : (Code to create table)

```
<? php
$user="username";
$password="password";
$database="database";
mysql_connect(localhost,$user,$password);
@mysql_select_db($database) or die( "Unable to select database");
$query="CREATE TABLE contacts (id int(6) NOT NULL,fname
varchar(15) NOT NULL,lname varchar(15) NOT NULL,phone
varchar(20) NOT NULL,mobile varchar(20) NOT NULL,fax
varchar(20) NOT NULL,email varchar(30) NOT NULL,web
varchar(30) NOT NULL,PRIMARY KEY (id),UNIQUE id (id),KEY
id_2 (id))";
mysql_query($query);
mysql_close();
?>
```

Managing Database in MySQL

- Create Database :-
  - CREATE DATABASE [IF NOT EXISTS] database_name;
  - E.g.:- CREATE DATABASE classicmodels;
- Showing Database:-Shows all databases in your database server.
  - SHOW DATABASES;
- Selecting Database to work with:-
  - USE database_name;
  - E.g. :- USE classicmodels;
- Removing Database :-
  - DROP DATABASE [IF EXISTS] database_name;
  - DROP DATABASE IF EXISTS temp_database;

Working With Tables

- Creating Table
  - CREATE TABLE [IF NOT EXISTS] table_name(column_list) type=table_type
  - Column_list – col_name datatype(size)
- Showing and Describing Tables in database
  - SHOW TABLES;
  - DESCRIBE table_name;
- Remove Table from the database
  - DROP [TEMPORARY] TABLE [IF EXISTS] table_name [, table_name,...]
  - TEMPORARY – is used to delete temporary tables from the database.

Working with Tables

- Empty Table's data faster and reset all auto-increment fields
- TRUNCATE TABLE table_name;

Managing Database Indexes

- Creating Indexes:-

  CREATE [UNIQUE|FULLTEXT] INDEX index_name USING [BTREE | HASH | RTREE]

  ON table_name (column_name [(length)] [ASC | DESC],...)
- Eg:- CREATE INDEX officeCode ON employees(officeCode);
- Removing Indexes:-
  - DROP INDEX index_name ON table_name
  - DROP INDEX officeCode ON employees

MySQL Select Statement

- SELECT column_name1,column_name2...

  FROM tables [WHERE conditions]

  [GROUP BY group [HAVING group_conditions]][ORDER BY sort_columns] [LIMIT limits];
- SELECT DISTINCT ColumnName from Tablename;
- SELECT col_1,col_2, col_3

  FROM tablename ORDER BY col_1 ASC, col_2 DESC;

How to Use MySQL Limit to Constrain Number of Returned Records

- SELECT * FROM table LIMIT N;
- SELECT columns FROM table LIMIT S, N;
- SELECT column_list FROM table_name WHERE column IN ("list_item1","list_item2"…);

SQL IN, BETWEEN Operator

- SELECT column_list FROM table_name

WHERE column IN ("list_item1","list_item2"…)

- SELECT officeCode, city, phone FROM offices WHERE country = 'USA' OR country = 'France'
- SELECT column_list FROM table_name WHERE column_1 BETWEEN lower_range AND upper_range
- SELECT column_list FROM table_name WHERE column_1 >= lower_range AND column_1 <= upper_range

How to Use MySQL LIKE to Select Data Based on Patterns Matching

- Percentage (%) wildcard allows you to match any string of zero or more characters
- Underscore (_) allows you to match any single character.
- SELECT employeeNumber, lastName, firstName FROM employees WHERE firstName LIKE 'a%‘;
- SELECT col1_,col_2,... col_n, aggregate_function(expression) FROM table WHERE where_conditions GROUP BY col_1, col_2, ... col_n ORDER BY column_list
- SELECT status FROM orders GROUP BY status;