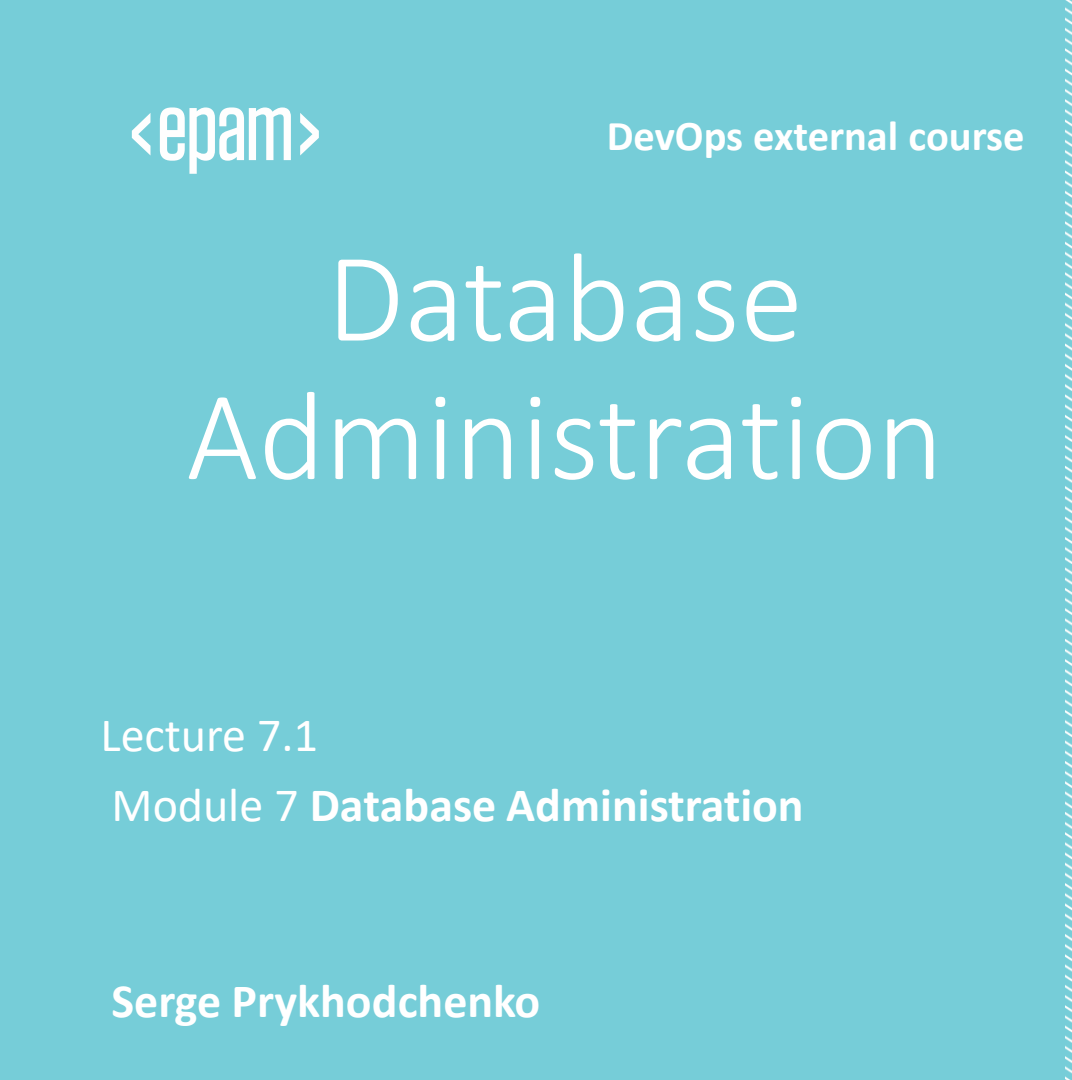


DevOps external course

Database Administration

Lecture 7.1

Module 7 **Database Administration**

Serge Prykhodchenko

Agenda

- Databases
- MySQL
- Q&A

DATABASES

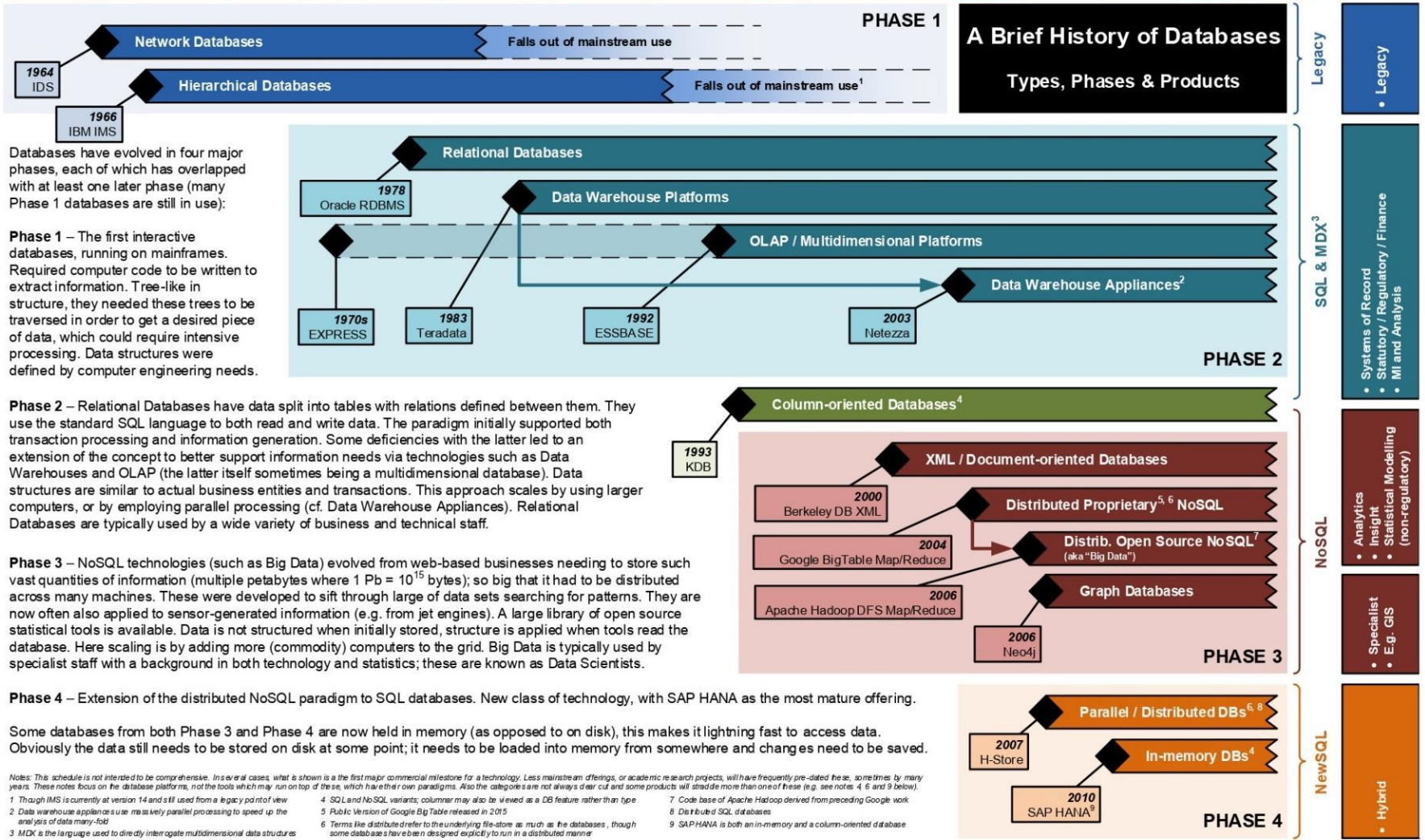
What is a database?

- A database is one of the important components for many applications and is used for storing a series of data in a single set. In other words, it is a group/package of information that is put in order so that it can be easily accessed, manage and update.

- A database is a persistent repository of data stored in a computer. The data represent recorded information. By persistent we mean that the data remain available indefinitely, after the software applications that use or create the data are closed, and even when the computer systems on which the data are stored reboot or crash due to software or hardware failures.

A Brief History of Databases

Types, Phases & Products



Databases have evolved in four major phases, each of which has overlapped with at least one later phase (many Phase 1 databases are still in use):

Phase 1 – The first interactive databases, running on mainframes. Required computer code to be written to extract information. Tree-like in structure, they needed these trees to be traversed in order to get a desired piece of data, which could require intensive processing. Data structures were defined by computer engineering needs.

Phase 2 – Relational Databases have data split into tables with relations defined between them. They use the standard SQL language to both read and write data. The paradigm initially supported both transaction processing and information generation. Some deficiencies with the latter led to an extension of the concept to better support information needs via technologies such as Data Warehouses and OLAP (the latter itself sometimes being a multidimensional database). Data structures are similar to actual business entities and transactions. This approach scales by using larger computers, or by employing parallel processing (cf. Data Warehouse Appliances). Relational Databases are typically used by a wide variety of business and technical staff.

Phase 3 – NoSQL technologies (such as Big Data) evolved from web-based businesses needing to store such vast quantities of information (multiple petabytes where 1 Pb = 10^{15} bytes); so big that it had to be distributed across many machines. These were developed to sift through large of data sets searching for patterns. They are now often also applied to sensor-generated information (e.g. from jet engines). A large library of open source statistical tools is available. Data is not structured when initially stored, structure is applied when tools read the database. Here scaling is by adding more (commodity) computers to the grid. Big Data is typically used by specialist staff with a background in both technology and statistics; these are known as Data Scientists.

Phase 4 – Extension of the distributed NoSQL paradigm to SQL databases. New class of technology, with SAP HANA as the most mature offering.

Some databases from both Phase 3 and Phase 4 are now held in memory (as opposed to on disk), this makes it lightning fast to access data. Obviously the data still needs to be stored on disk at some point; it needs to be loaded into memory from somewhere and changes need to be saved.

Notes: This schedule is not intended to be comprehensive. In several cases, what is shown is a the first major commercial milestone for a technology. Less mainstream offerings, or academic research projects, will have frequently pre-dated these, sometimes by many years. These notes focus on the database platforms, not the tools which may run on top of these, which have their own paradigms. Also the categories are not always clear cut and some products will straddle more than one of these (e.g. see notes 4, 6 and 9 below).

1 Though IMS is currently at version 14 and still used from a legacy point of view
2 Data warehouse appliances use massively parallel processing to speed up the analysis of data many-fold
3 MDX is the language used to directly interrogate multidimensional data structures
4 SQL and NoSQL variants; columnar may also be viewed as a DB feature rather than type
5 Public Version of Google BigTable released in 2015
6 Terms like distributed refer to the underlying file-store as much as the databases, though some databases have been designed explicitly to run in a distributed manner
7 Code base of Apache Hadoop derived from preceding Google work
8 Distributed SQL databases
9 SAP HANA is both an in-memory and a column-oriented database

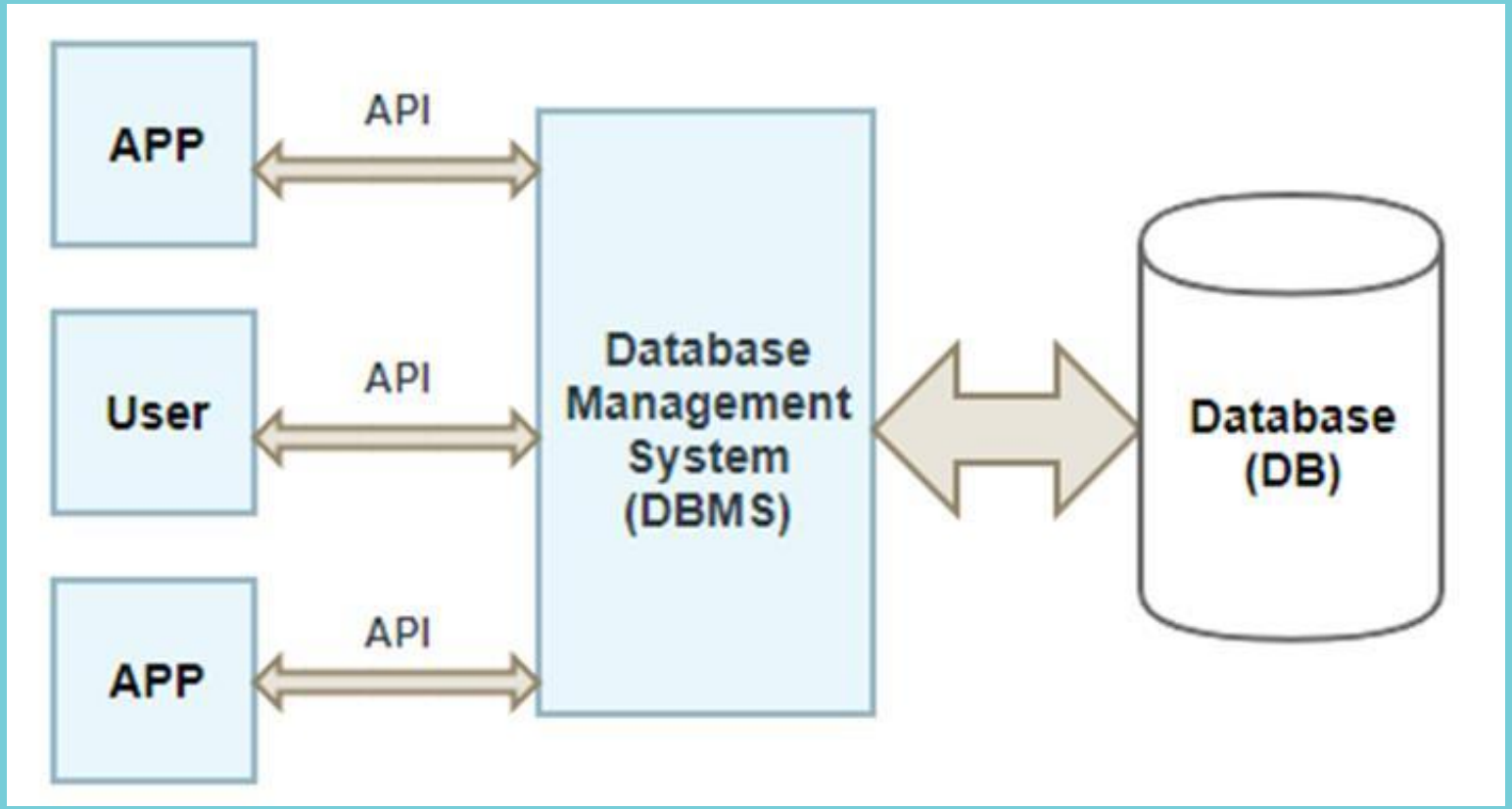
Acknowledgements:
a. Neil Raden (@neilraden) from HiredBrains (hiredbrains.com) – For both reviewing earlier drafts and providing significant input

Evolution of databases

- NewSQL
- NoSQL
- Object-oriented
- Relational
- Network
- File system

https://uk.wikipedia.org/wiki/NewSQL

Database Management System



Characteristics of DBMS

- It uses a digital repository established on a server to store and manage the information.
- It can provide a clear and logical view of the process that manipulates data.
- DBMS contains automatic backup and recovery procedures.
- It contains ACID properties which maintain data in a healthy state in case of failure.
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data.
- It is used to provide security of data.
- It can view the database from different viewpoints according to the requirements of the user.

Advantages of DBMS

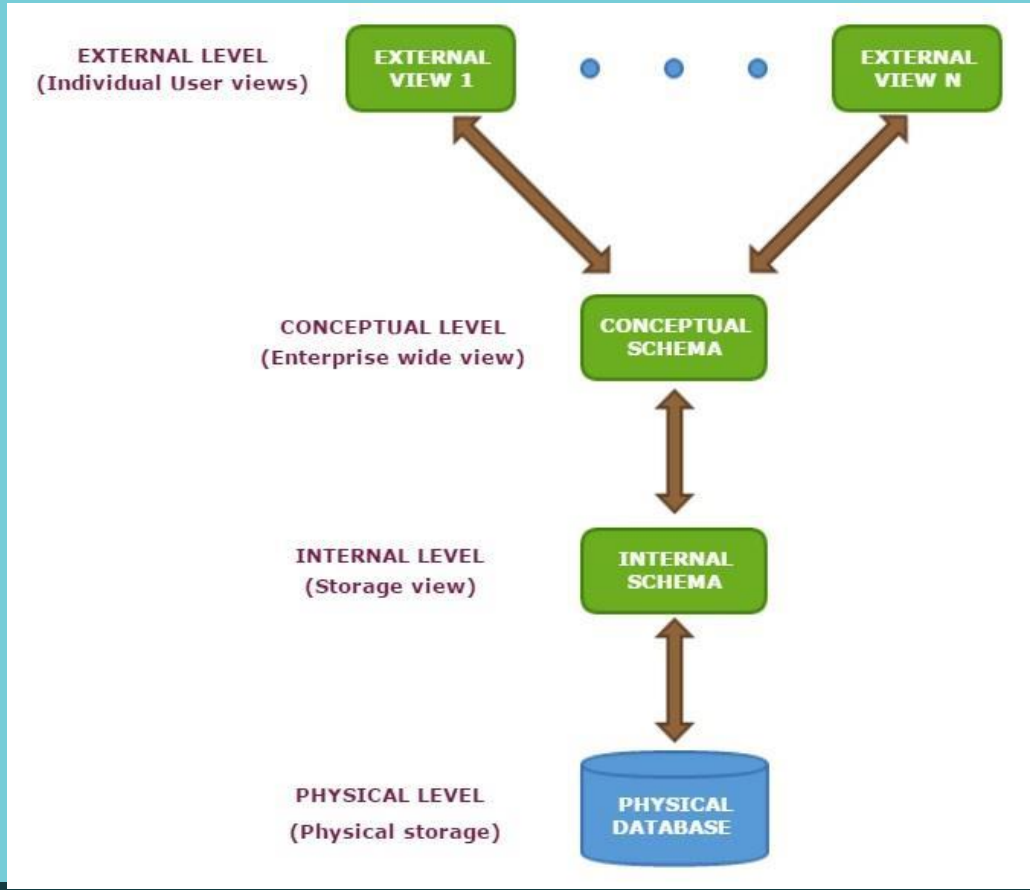
- **Controls database redundancy**: It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
- **Data sharing**: In DBMS, the authorized users of an organization can share the data among multiple users.
- **Easily Maintenance**: It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time**: It reduces development time and maintenance need.
- **Backup**: It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **Multiple user interface**: It provides different types of user interfaces like graphical user interfaces, application program interfaces

Disadvantages of DBMS

- **Cost of Hardware and Software**: It requires a high speed of data processor and large memory size to run DBMS software.
- **Size**: It occupies a large space of disks and large memory to run them efficiently.
- **Complexity**: Database system creates additional complexity and requirements.
- **Higher impact of failure**: Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

Database Architecture

Three Level Database Architecture



Internal Level

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

Conceptual Level

- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level

External Level

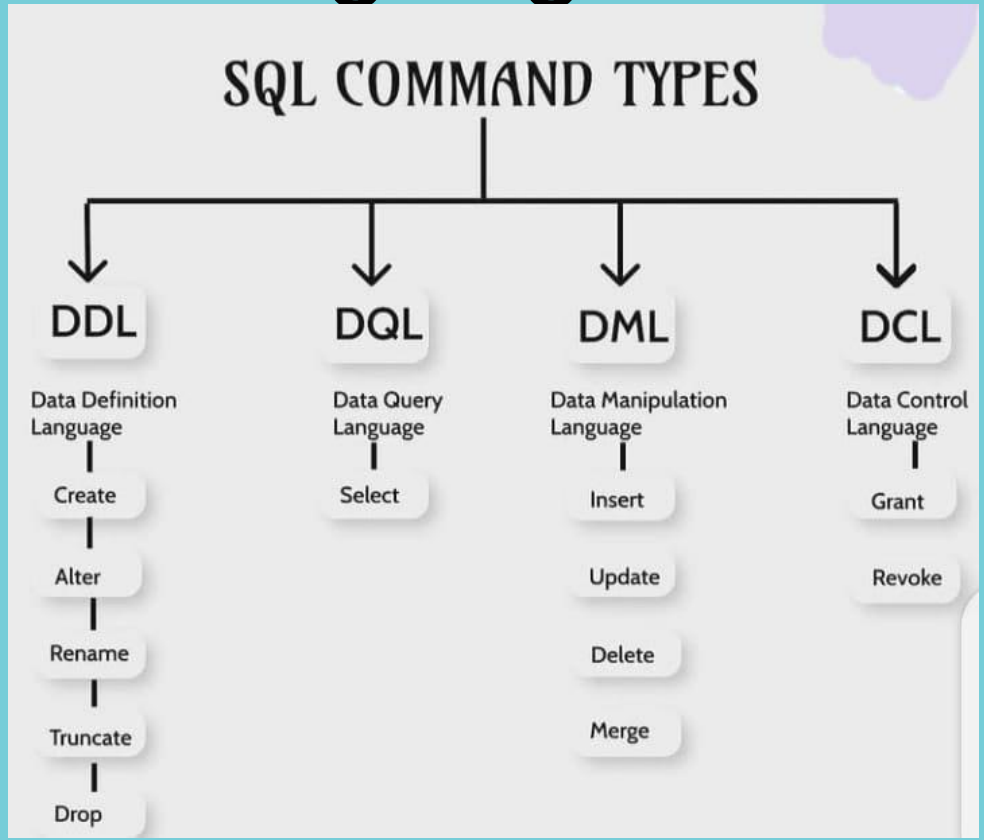
- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems

Data model Schema and Instance

- The data which is stored in the database at a particular moment of time is called an instance of the database.
- The overall design of a database is called schema.
- A database schema is the skeleton structure of the database. It represents the logical view of the entire database.
- A schema contains schema objects like table, foreign key, primary key, views, columns, data types, stored procedure, etc.
- A database schema can be represented by using the visual diagram. That diagram shows the database objects and relationship with each other.
- A database schema is designed by the database designers to help programmers whose software will interact with the database. The process of database creation is called data modeling.

Types of Database Language

- DDL - Data Definition Language. It is used to define database structure or pattern
- DML - Data Manipulation Language. It is used for accessing and manipulating data in a database
- DCL - Data Control Language. It is used to retrieve the stored or saved data
- TCL - Transaction Control Language is used to run the changes made by the DML statement



Data Definition Language

- It is used to create schema, tables, indexes, constraints, etc. in the database.
- Using the DDL statements, you can create the skeleton of the database.
- Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Here are some tasks that come under DDL:

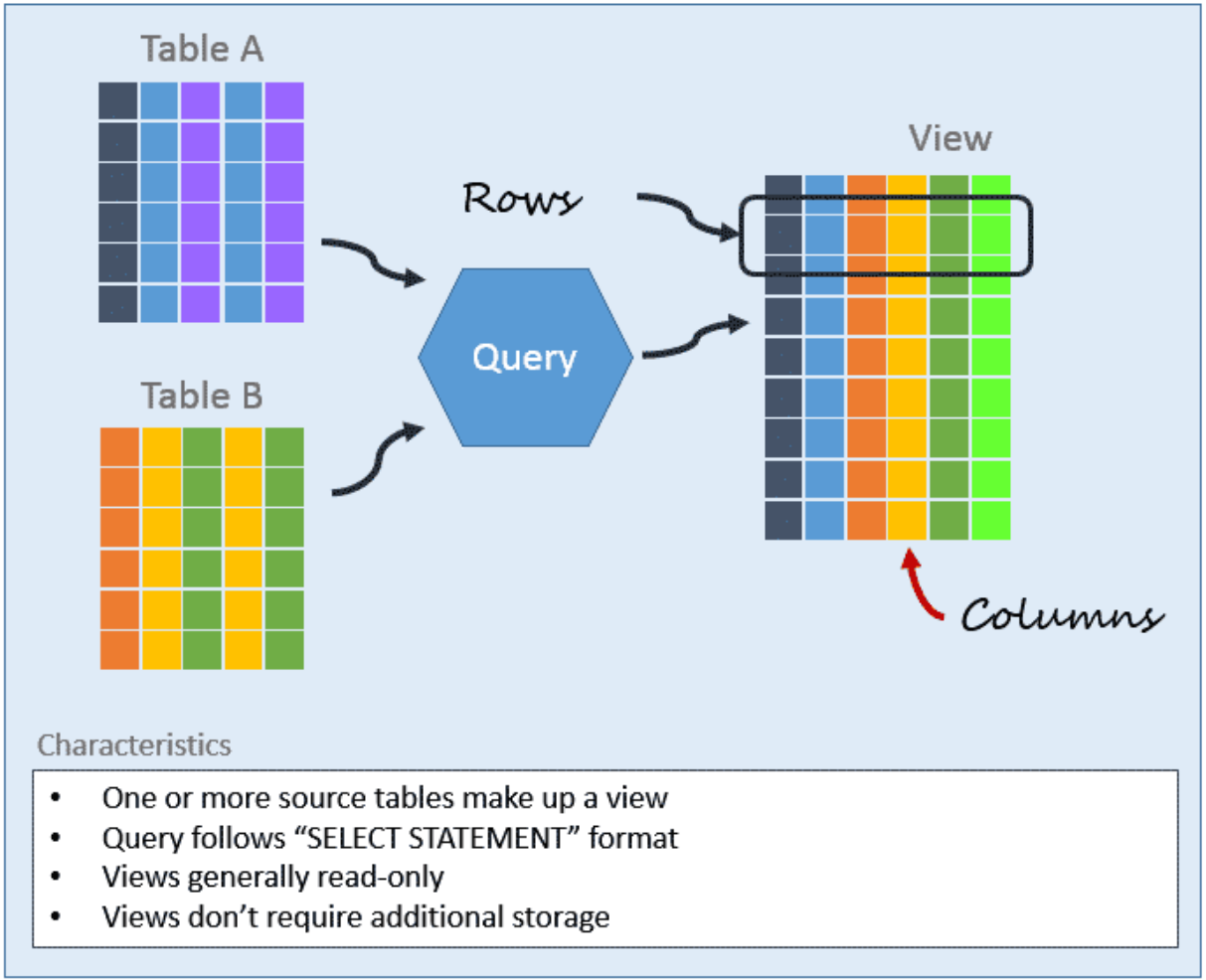
- **Create**: It is used to create objects in the database.
- **Alter**: It is used to alter the structure of the database.
- **Drop**: It is used to delete objects from the database.
- **Truncate**: It is used to remove all records from a table.
- **Rename**: It is used to rename an object.
- **Comment**: It is used to comment on the data dictionary.

Data Manipulation Language

Here are some tasks that come under DML:

- Select: It is used to retrieve data from a database.
- Insert: It is used to insert data into a table.
- Update: It is used to update existing data within a table.
- Delete: It is used to delete all records from a table.
- Merge: It performs UPSERT operation, i.e., insert or update operations.
- Call: It is used to call a structured query language or a Java subprogram.
- Explain Plan: It has the parameter of explaining data.
- Lock Table: It controls concurrency.

Anatomy of a View



Characteristics

- One or more source tables make up a view
- Query follows “SELECT STATEMENT” format
- Views generally read-only
- Views don’t require additional storage

Data Control Language and Transaction Control Language

Here are some tasks that come under DCL:

- Grant: It is used to give user access privileges to a database.
- Revoke: It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:

CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

TCL can be grouped into a logical transaction.

Here are some tasks that come under TCL:

- Commit: It is used to save the transaction on the database.
- Rollback: It is used to restore the database to original since the last Commit.

ER model

• ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
• It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
• In ER modeling, the database structure is portrayed as a diagram called an entity relationship diagram.
- An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.
- A relationship is used to describe the relation between entities.

MYSQL

MySQL Introduction

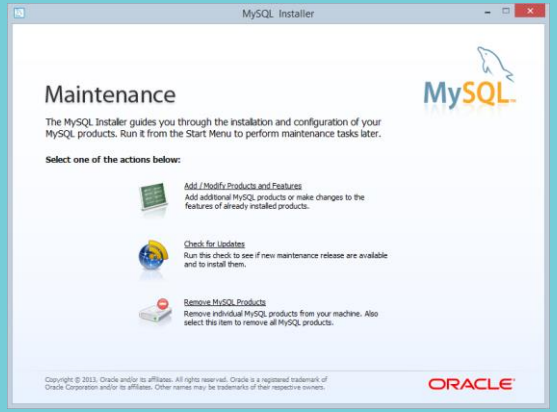
- Open Source (C/C++), Free
- High Performance, Low Cost, High Reliability
- LAMP (Linux+Apache+MySQL+PHP)
- Multi-OS Support (Windows, Linux, MacOS)
- API Support (C/C++, C#, Python, PHP, Java)

What is MySQL?

- MySQL Database Server
  - MySQL is open-source DB server (RDBMS)
  - World's most-popular open-source database
  - Mostly used to power web sites and small apps
  - Supports concurrency, transactions (full ACID)
  - Stored procedures, views, triggers, partitioning
  - Support clustering and replication
- Free and paid editions
  - Community Server, Enterprise, Cluster CGE

MySQL Community Server

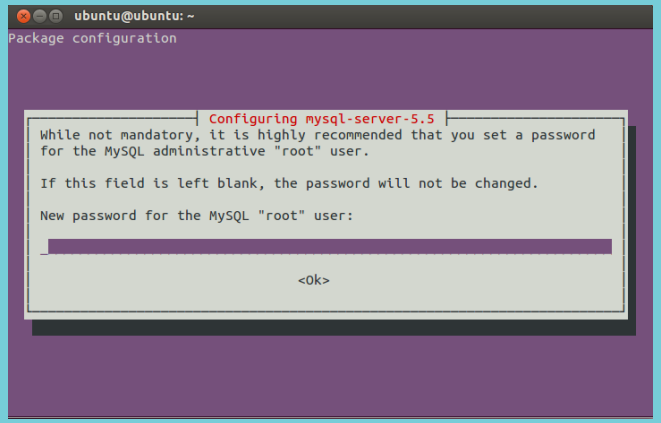
- MySQL Community Server
  - The free open-source MySQL edition
  - MySQL for Windows:
    - Pre-packaged installer available from http://dev.mysql.com/downloads/mysql/
  - MySQL for Linux:
    - Available through the package managers




```
sudo apt-get install mysql-server
```
(Debian / Ubuntu)

```
sudo yum install mysql-server
```
(Red Hat / CentOS)

MySQL Installation



- OS: Ubuntu Linux
- Official Site: http://dev.mysql.com/downloads/mysql/
  - Select Platform and download a deb package
  - Latest version: MySQL Community Server
- Using apt-get (Recommended)
  - sudo apt-get install mysql-server mysql-client
  - Set a password for root account
- For Windows, don’t forget to set Windows system PATH environment variable

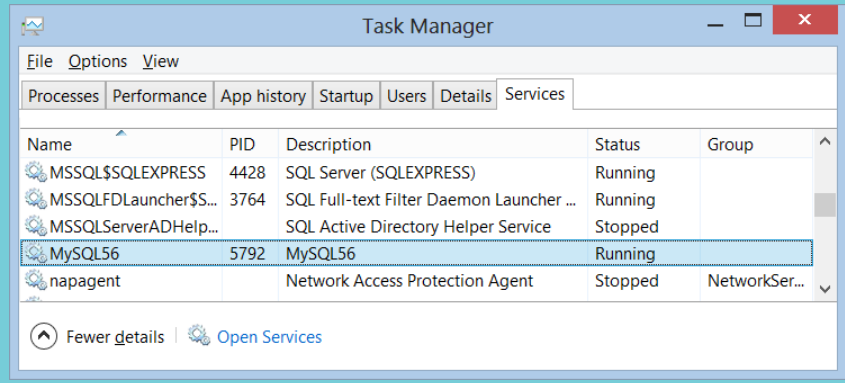
MySQL root password change

- mysql –version
- sudo systemctl stop mysql.service
- sudo systemctl status mysql.service
- sudo systemctl set-environment MYSQLD_OPTS="--skip-networking --skip-grant-tables“
- sudo systemctl start mysql.service
- sudo systemctl status mysql.service
- sudo mysql -u root
- *flush privileges;*
- *USE mysql*
- *ALTER USER 'root'@'localhost' IDENTIFIED BY 'the-new-password’;*
- *quit;*
- sudo systemctl unset-environment MYSQLD_OPTS
- sudo systemctl revert mysql
- killall -u mysql
- sudo systemctl restart mysql.service

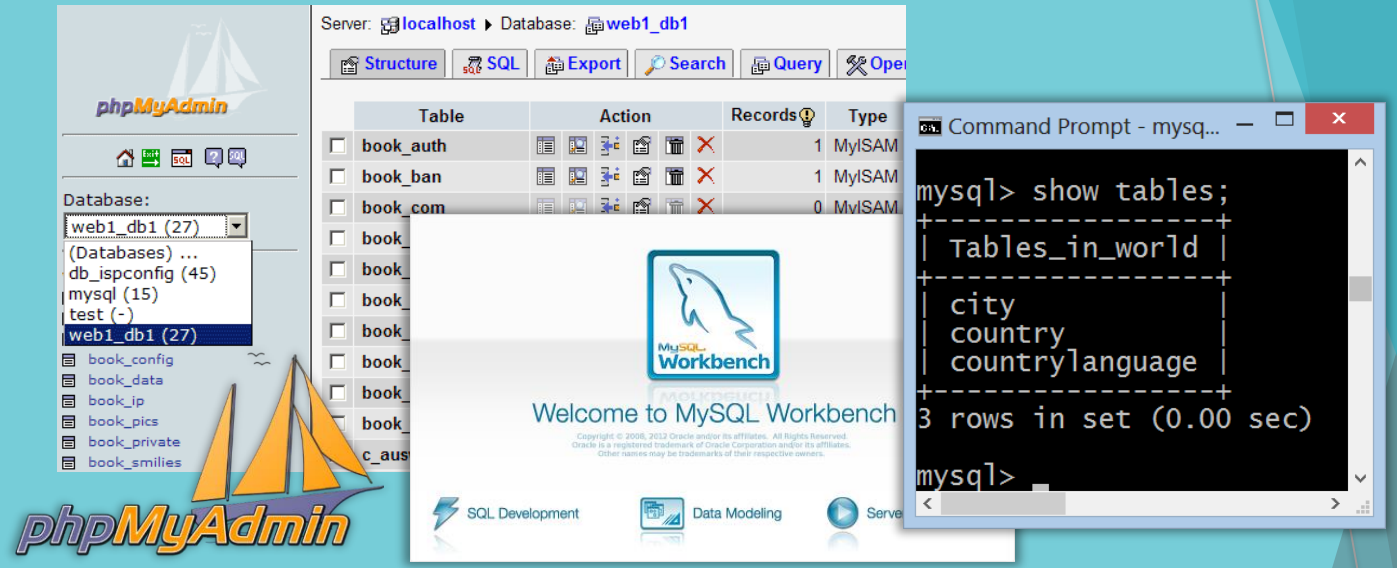
https://linuxhint.com/change-mysql-root-password-ubuntu/

MySQL Services, Start, Stop

- MySQL services in Windows
  - Just one service: **MySQLXX**
  - Starting: `net start MySQL56`
  - Stopping: `net stop MySQL56`



- MySQL services in Linux
  - Starting: `sudo service mysql start` `sudo systemctl start mysql.service`
  - Stopping: `sudo service mysql stop` `sudo systemctl stop mysql.service`

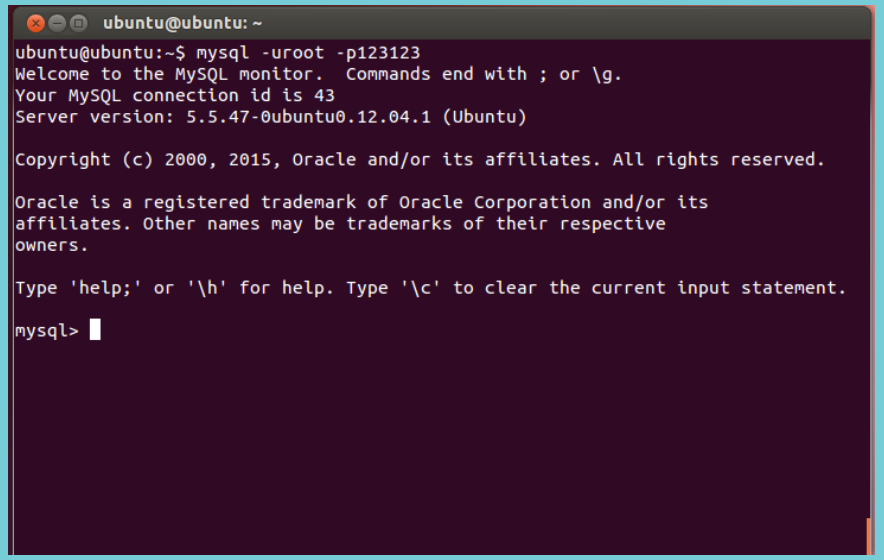


MySQL Administration Tools

The Console MySQL Client, MySQL Workbench, phpMyAdmin

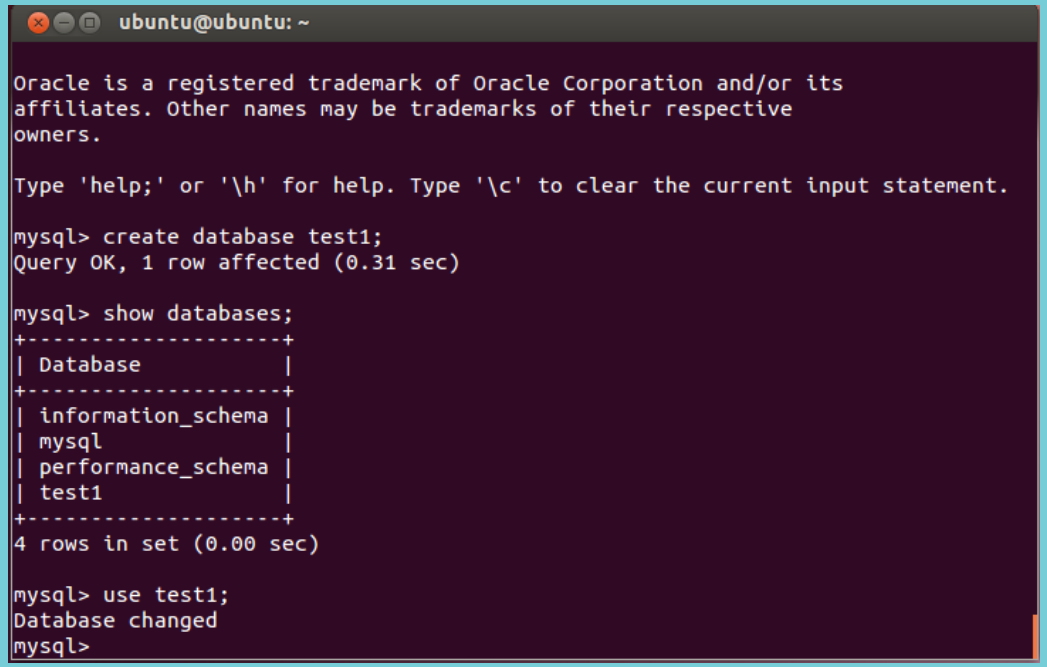
MySQL Command-Line Tool

- Login
  - mysql –u<username> –p<password>
  - mysql –u root –p 123123
  - mysql -u root - p
- Logout
  - exit



MySQL Command-Line Tool

- Input your SQL after “mysql>”



Creating Databases and Tables

```
CREATE DATABASE books
```

- Creating a database
- Creating tables

```
USE books;

CREATE TABLE authors (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,
  PRIMARY KEY (id)
);
```

```
CREATE TABLE books (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(150) NOT NULL,
  isbn VARCHAR(13) NULL,
  PRIMARY KEY (id)
);
```

MySQL Command-Line Tool

```
mysql> create table mytable(data int);
Query OK, 0 rows affected (0.08 sec)

mysql> show tables;
+----------------+
| Tables_in_test1 |
+----------------+
| mytable        |
+----------------+
1 row in set (0.00 sec)

mysql> insert into mytable (data) values (1),(2),(3),(4);
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from mytable;
+------+
| data |
+------+
|    1 |
|    2 |
|    3 |
|    4 |
+------+
4 rows in set (0.00 sec)
```

```
mysql> select * from mytable where data > 2;
+------+
| data |
+------+
|    3 |
|    4 |
+------+
2 rows in set (0.00 sec)

mysql> delete from mytable where data = 3;
Query OK, 1 row affected (0.32 sec)

mysql> select * from mytable where data > 2;
+------+
| data |
+------+
|    4 |
+------+
1 row in set (0.00 sec)
```

Edit Tables and Table Data

- Altering tables

```
ALTER TABLE books ADD COLUMN author_id INT NULL AFTER isbn;

ALTER TABLE books ADD INDEX FK_books_authors_idx (author_id ASC);

ALTER TABLE books ADD CONSTRAINT FK_books_authors
FOREIGN KEY (author_id) REFERENCES authors (id);
```

- Inserting data rows

```
INSERT INTO authors (name) VALUES ('Nakov');

INSERT INTO books (name, author_id, isbn)
VALUES ('Intro C#', 1, '9789544005276');
```

DB example

```
create table students(
        sid int,
        name varchar(40),
        dept varchar(40),
        age int,
        primary key(sid)
);
```

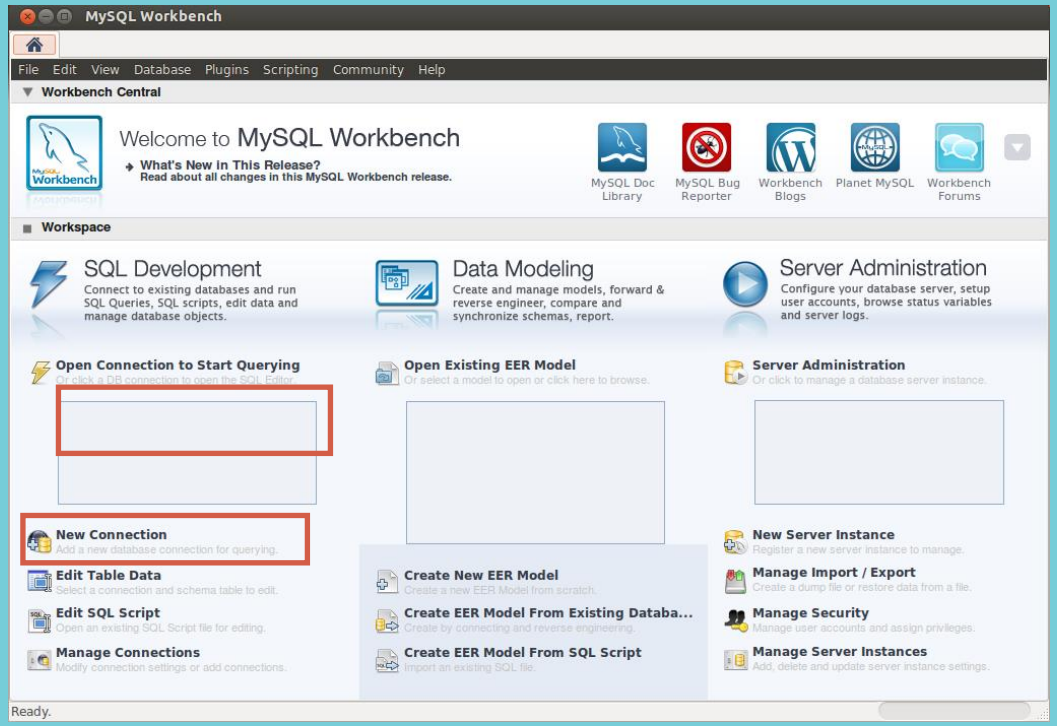
```
create table courses(
        cid int,
        cname varchar(40),
        spring int,
        teacher varchar(40),
        primary key(cid)
);
```

```
create table sc (
        sid int references students(sid) ON DELETE CASCADE ON UPDATE CASCADE,
        cid int ,
        semester int,
        cname varchar(40),
         grade int
);
```

MySQL Workbench Installation

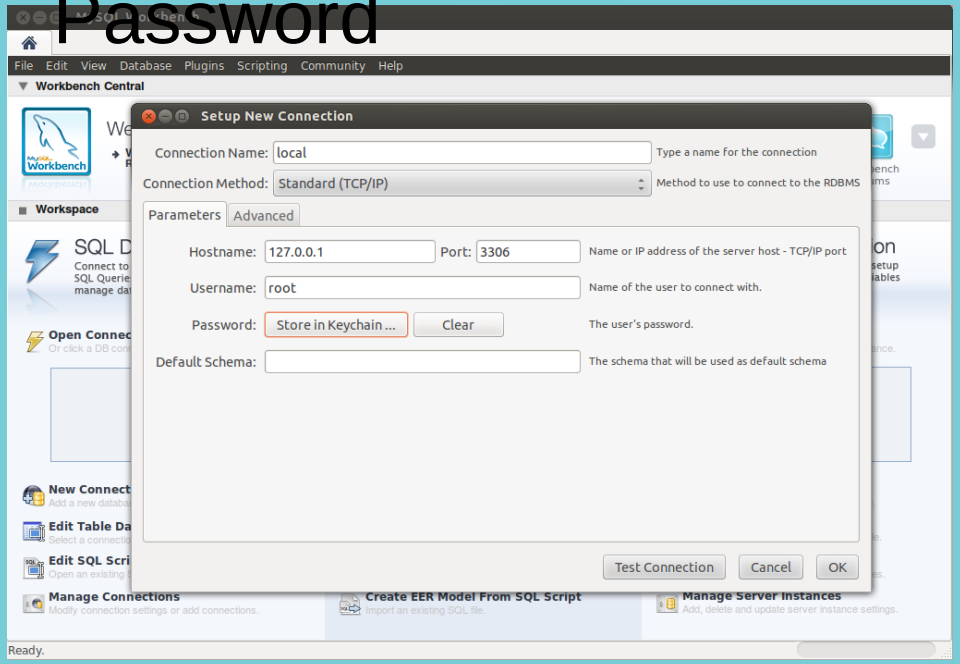
- A visual tool for MySQL
- Official Site: http://dev.mysql.com/downloads/workbench/
  - Select Platform and download a deb package
  - Latest version: MySQL Workbench 6.3.6
- Using apt-get (Recommanded)
  - sudo apt-get install mysql-workbench

MySQL Workbench Login

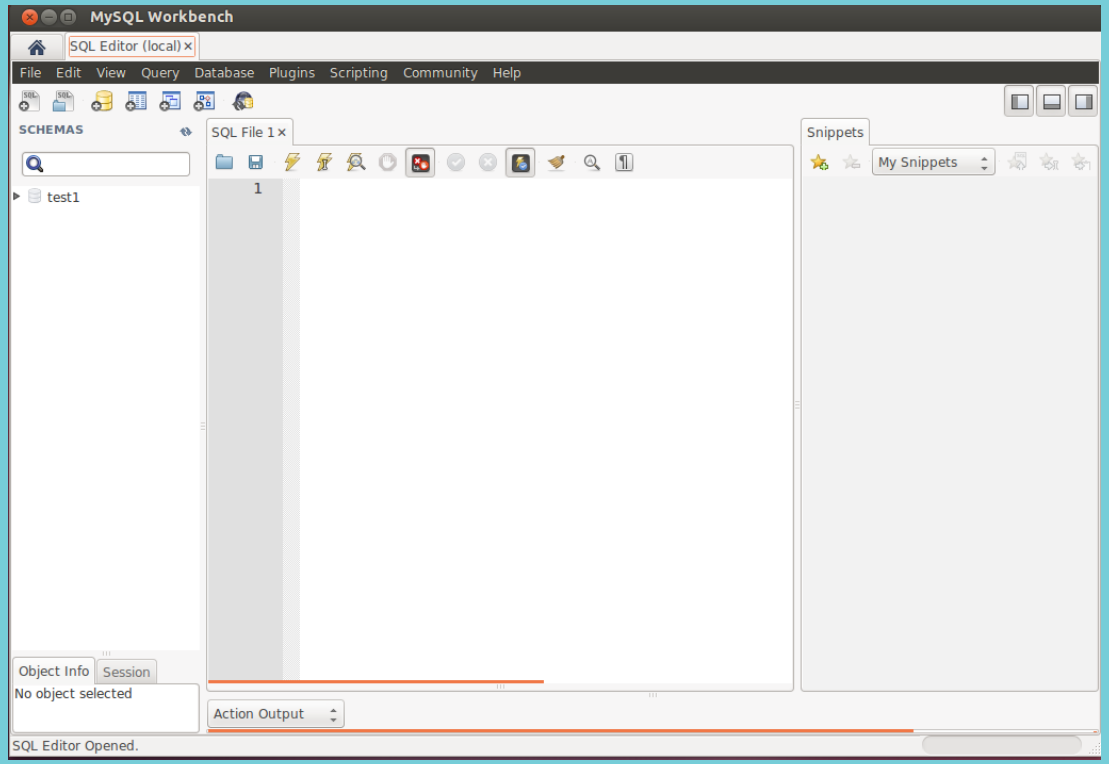


MySQL Workbench Login

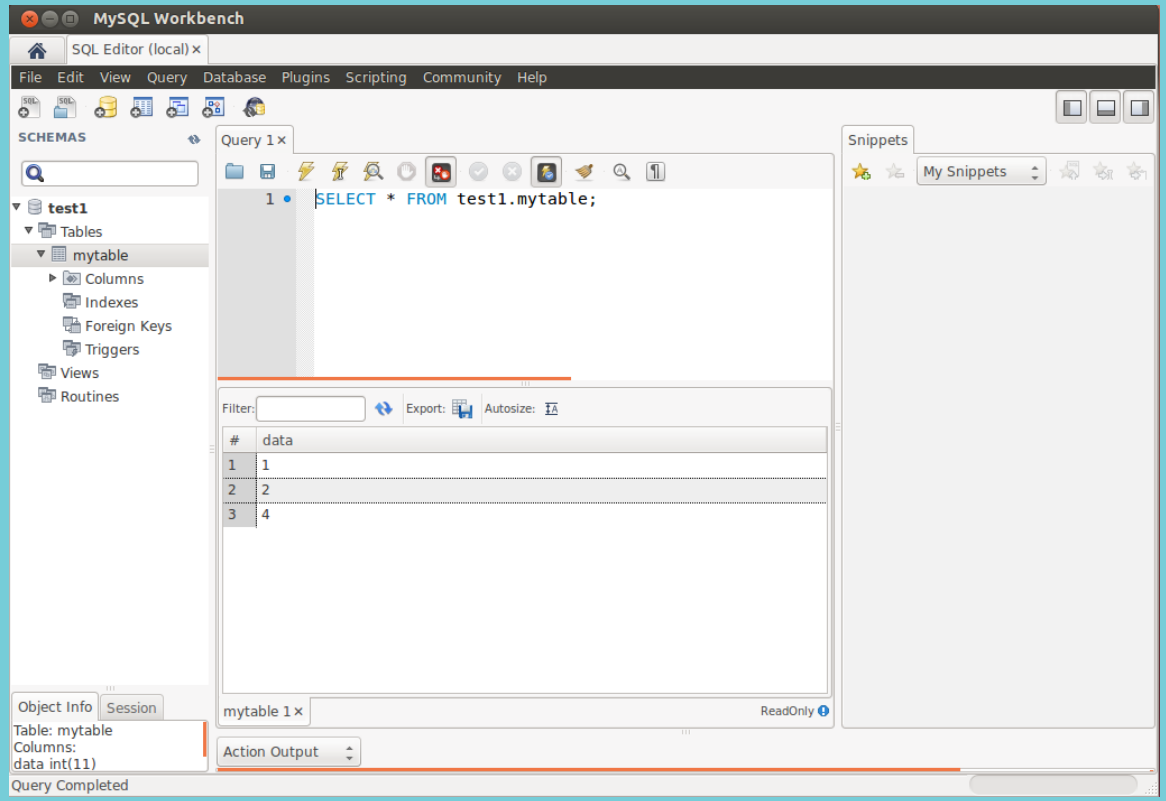
- Hostname, Port, Username, Password



MySQL Workbench UI

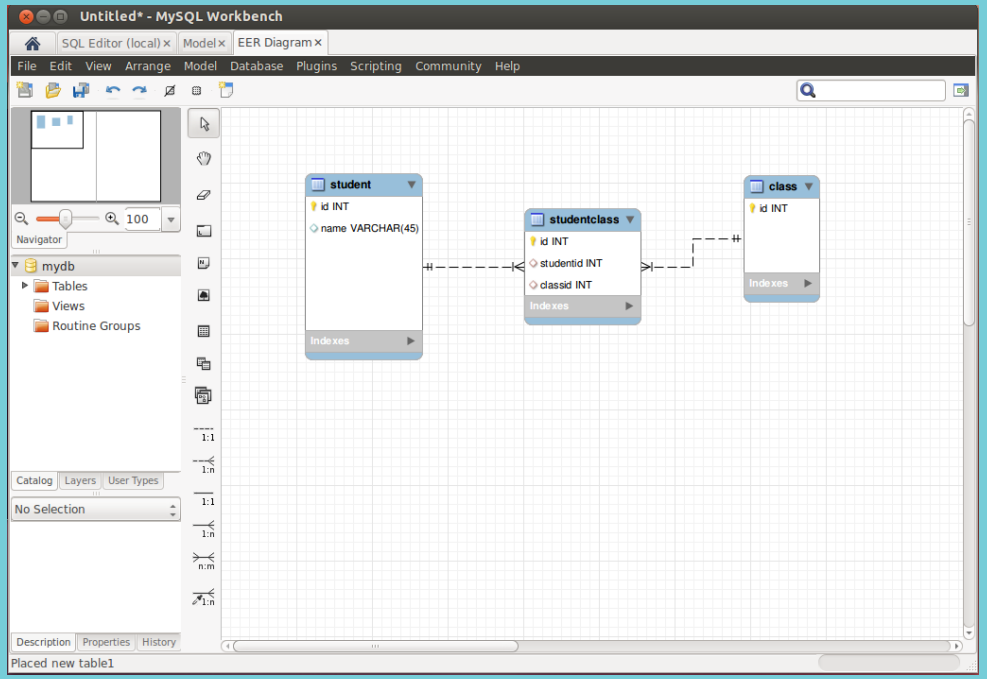


MySQL Workbench Usage



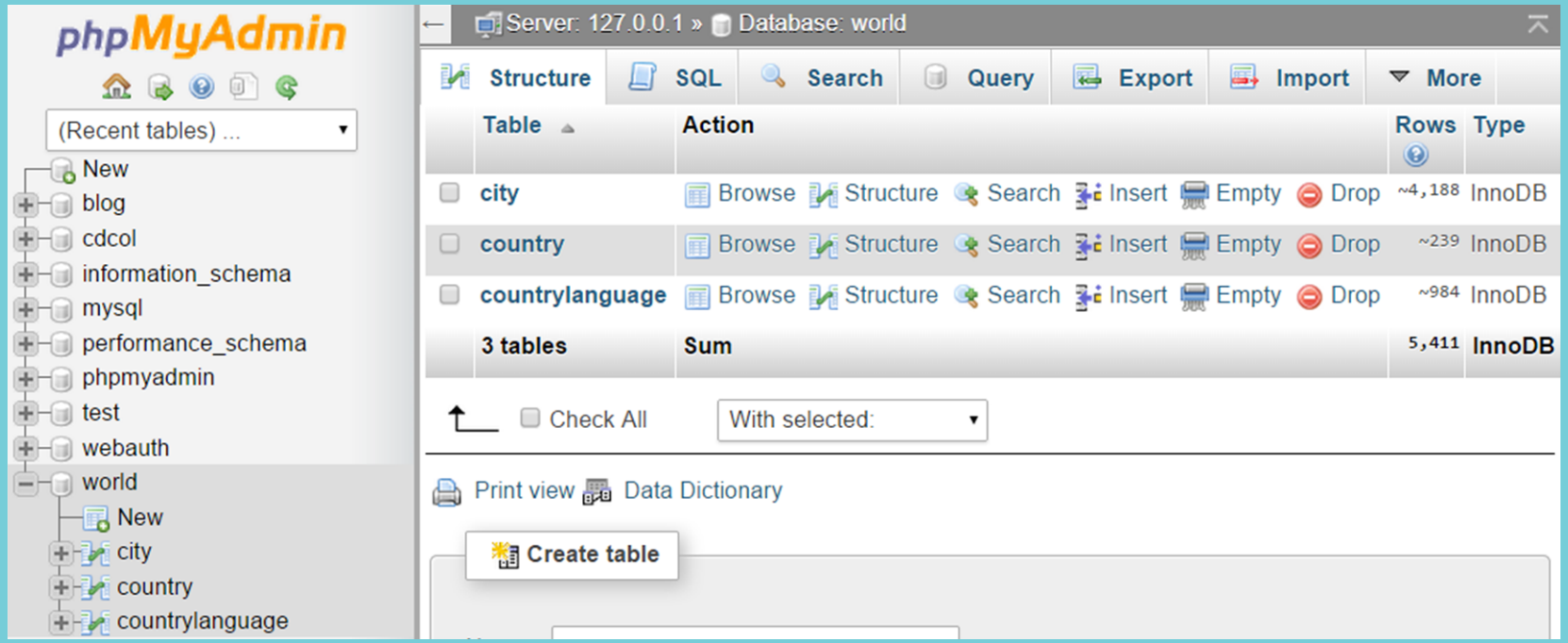
MySQL Workbench ER Model & Forward Engineering

- File->New Model->Add Diagram
- File->Export
- Get SQL Script



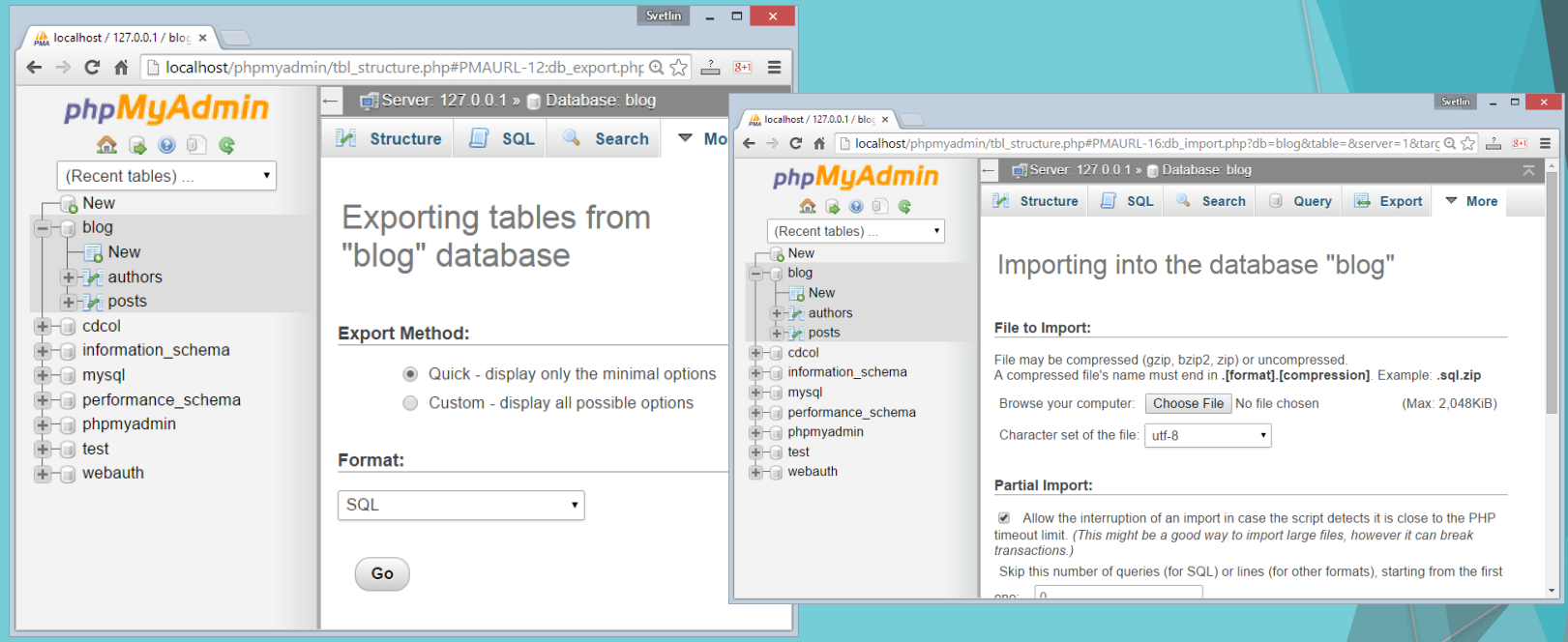
phpMyAdmin Tool

- phpMyAdmin – Web-based open-source MySQL admin tool



Moving a MySQL Database

- To move MySQL database to another server
  - Use the SQL export / SQL import features
- Export a database to SQL script
  - MySQL Workbench → Server Administration → Data Export → Export to Self-Contained File
  - phpMyAdmin → Export → SQL
- Import a database from SQL script
  - Just execute the script in Workbench
  - phpMyAdmin → Import → SQL



Import / Export MySQL Database

Other References

- https://en.wikipedia.org/wiki/MySQL
- http://www.runoob.com/mysql/mysql-install.html
- http://blog.csdn.net/ithomer/article/details/5131863
- http://www.runoob.com/python/python-mysql.html


THANK YOU!