

DATABASE SYSTEMS

Introduction to MySQL



Database System Course, 2018

CAUTION!



This class is NOT a recitation

We will NOT discuss the course material relevant to the exam and homework assignment

We have three full lectures:

- Introduction to MySQL
- Database programming
- Web programming

We will focus ONLY on the practical side of databases.

It is relevant for your **FINAL PROJECT (35% of your grade)**

ADMINISTRATION



Homework Submission

- Submission date is on our website.. (No late arrivals will be accepted)
- Work should be done in pairs
- Submission is done via Moodle, by one of the partners
- Submit a zip file, with
 - an answers pdf that contains the full names and IDs of both partners on top of the page
 - A .sql file for every query. **Make sure it's runnable.**

ADMINISTRATION



The Final Project

- You will develop a fully working website with a database.
- Focus on the database design, optimizations, SQL queries, and DB programming best practices.
- You will most definitely encounter and use elements which are NOT on the class/TA material.
- Work in groups of 4-5.

SUPPORT & QUESTIONS



System Support

- During this class, you will use several servers/frameworks belonging to the university e.g. **Nova**, **MySql Server**, and the **Python** web-server.
- If you encounter a system problem, you have to email **system@cs.tau.ac.il**. They are really nice and will help you if you encounter problems.
- Please do not email FIRST about such problems, but if things are not solved, I can help

SUPPORT & QUESTIONS



Technical/Programming Issues

- Remember that MySQL, SQL, Database programming and web programming are among **the most common topics in the computer science community**.
- Use Google, StackOverflow, watch online tutorials and video lectures.

SUPPORT & QUESTIONS



Other Issues

- Use the Moodle Forums for:
 - Finding a partner/team (this is your responsibility)
 - Homework/project questions - we will have a dedicated forum for each hw assignment.
- Please do not email us questions related to the hw/project unless its REALLY PERSONAL and may not concern other students
- Come to my office hours (by appointment)! I am much more available and communicative in person than in mail/forum. (Details on locations/times on my academic **website**.)

AGENDA FOR TODAY



~~Administration~~



Part #1: Database architecture on the web

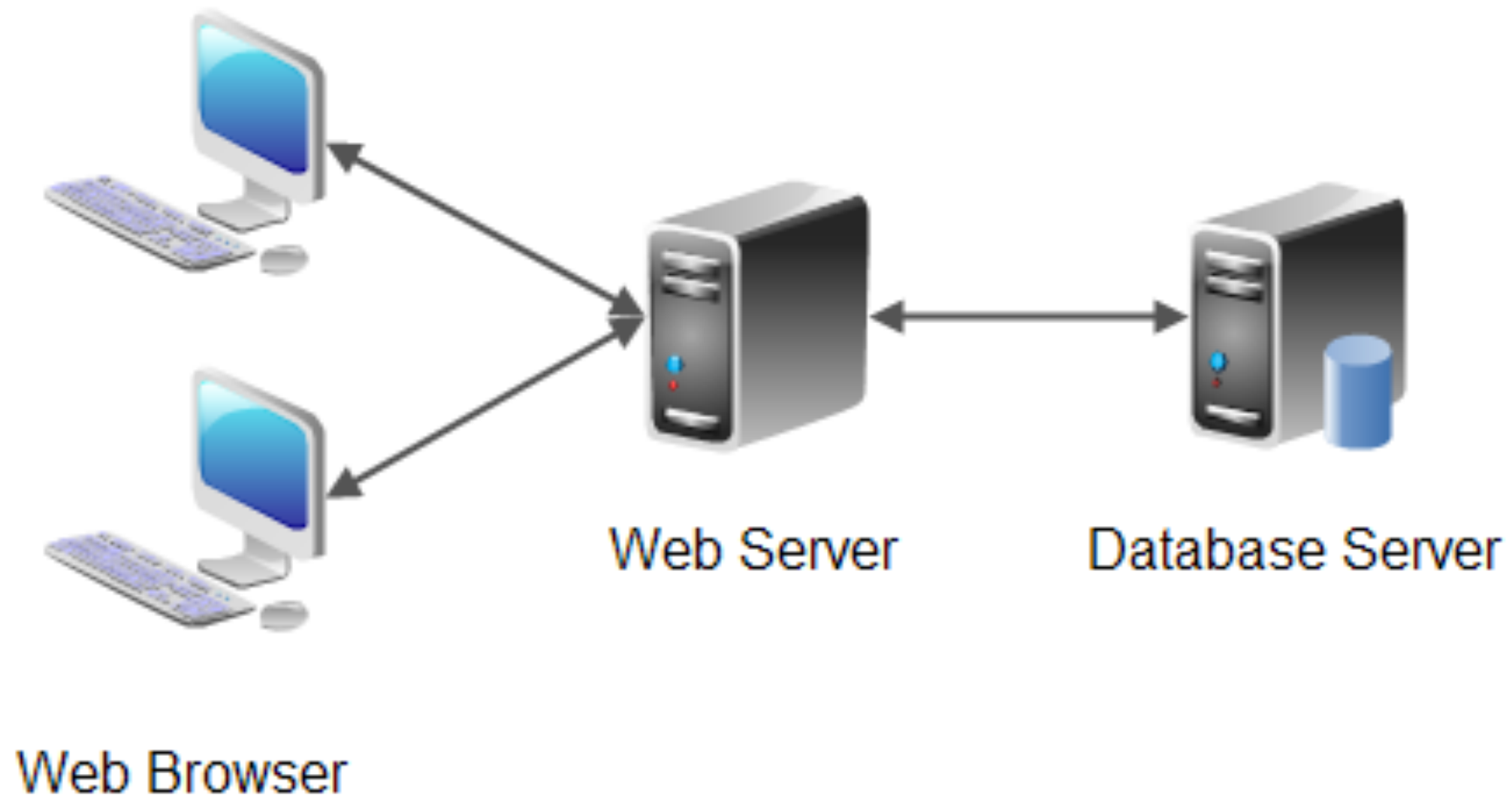


Part #2: Databases usage in the last 20K years



Part #3: Introduction to Mysql and how to do HW#1

DATABASE ARCHITECTURE ON THE WEB (ILLUSTRATION)



DATABASE ARCHITECTURE ON THE WEB (BRIEF)

 Database server is a **standalone** server.

 Database server is not accessible to web-users
(when configured securely)

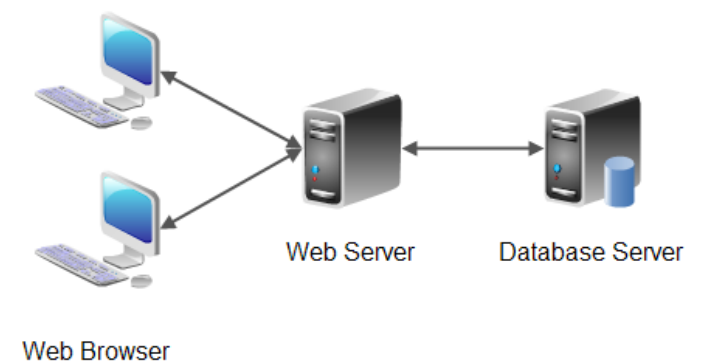
 Only the web server communicates with the DB.

 Administrators have special permissions to access
to the database management system directly.

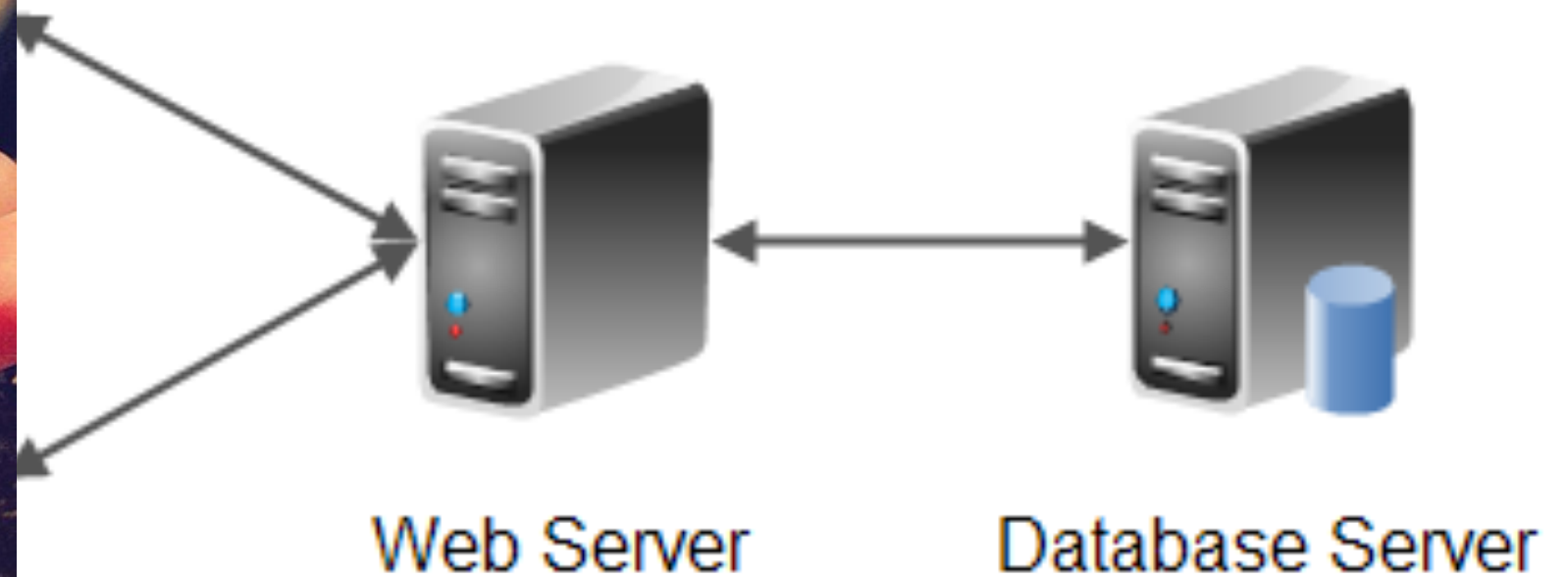
DATABASE ARCHITECTURE ON THE WEB (EXTENDED)

Web session illustration in 6 simple stages

1. A client opens a web browser in her computer
2. Within the web browser she type the URL of a website (e.g. ynet.co.il)
3. The browser issues an HTTP session to request the website's content.
4. The web server receive the HTTP request
5. The web server connects to the DB server to retrieve data (e.g., current articles of today)
6. The web server returns the client the content of the page.



HOW DOES INSTAGRAM WORKS?



HOW DOES INSTAGRAM WORKS?



Web Server

Database Server

- ★ Image Processing
- ★ UI operations

- ★ Authentication
- ★ Notifications
- ★ API

- ★ Images table
- ★ Users table

AGENDA FOR TODAY



~~Administration~~



~~Part #1: Database architecture on the web~~



Part #2: Databases usage in the last 20K years



Part #3: Introduction to Mysql and how to do HW#1

DATABASE HISTORY

1966 IBM: Information Management System

 Designed for the Apollo space program, to store inventory, components and materials for Saturn V rocket. It was running on an IBM mainframe computer.

 IMS was a **hierarchical database**, relying on the "manual" navigation of a linked data set which was formed into a large network. Applications could find records by one of three methods:

1. Use of a primary key (known as a CALC key, typically implemented by hashing)
2. Navigating relationships (called sets) from one record to another
3. Scanning all the records in a sequential order



DATABASE HISTORY

1970 The relational model (theoretical)

 Mechanical hard drives invented

 It's sucks to search in the hierarchical DB,

 Invented by Edgar Codd from IBM

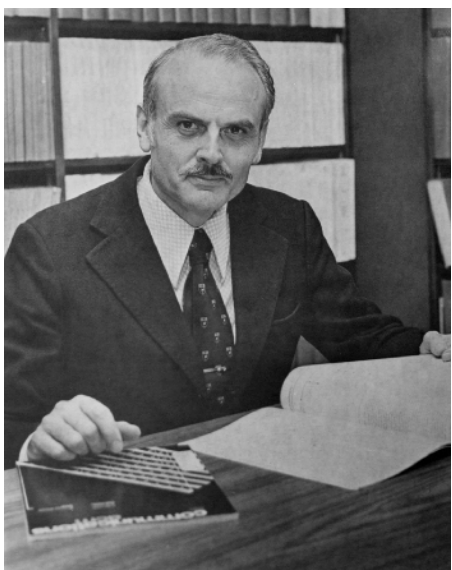
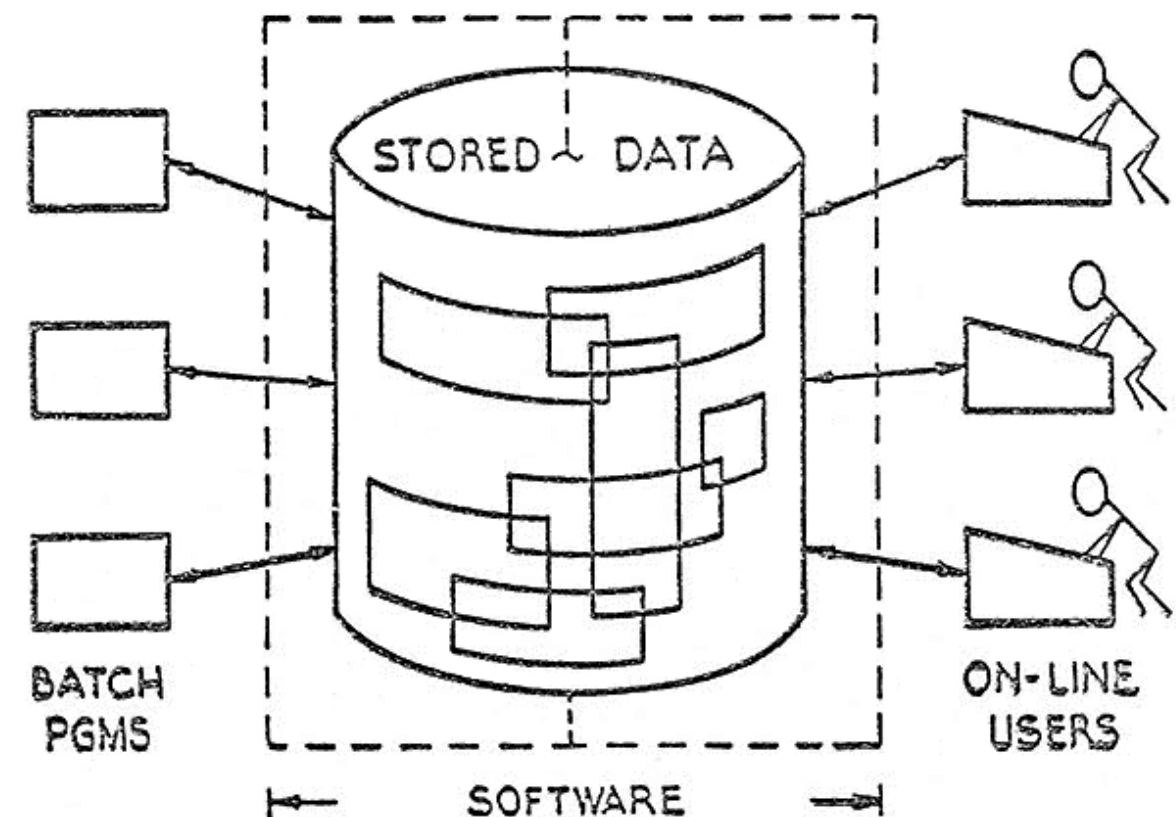
1974 IBM “System R”

 R is for relational.

 First implementation of SQL

 Proving the performance and usability of the relational model

A DATABASE SYSTEM



DATABASE HISTORY

1980 Personal Databases

 Desktops are introduced to the world

 People use spread-sheet software Like IBM Lotus

A:A1: 'EMP' MENU

Worksheet		Range	Copy	Move	File	Print	Graph	Data	System	Quit
Global		Insert	Delete	Column	Erase	Titles	Window	Status	Page	Hide
A	A	B	C	D	E	F	G			
1	EMP	EMP NAME	DEPTNO	JOB	YEARS	SALARY	BONUS			
2	1777	Azibad	4000	Sales	2	40000	10000			
3	81964	Brown	6000	Sales	3	45000	10000			
4	40370	Burns	6000	Mgr	4	75000	25000			
5	50706	Caesar	7000	Mgr	3	65000	25000			
6	49692	Curly	3000	Mgr	5	65000	20000			
7	34791	Dabarrett	7000	Sales	2	45000	10000			
8	84984	Daniels	1000	President	8	150000	100000			
9	59937	Dempsey	3000	Sales	3	40000	10000			
10	51515	Donovan	3000	Sales	2	30000	5000			
11	48338	Fields	4000	Mgr	5	70000	25000			
12	91574	Fiklore	1000	Admin	8	35000	---			
13	64596	Fine	5000	Mgr	3	75000	25000			
14	13729	Green	1000	Mgr	5	90000	25000			
15	55957	Hermann	4000	Sales	4	50000	10000			
16	31619	Hodgedon	5000	Sales	2	40000	10000			
17	1773	Howard	2000	Mgr	3	80000	25000			
18	2165	Hugh	1000	Admin	5	30000	---			
19	23907	Johnson	1000	VP	1	100000	50000			
20	7166	Laflare	2000	Sales	2	35000	5000			

DATA.WK3



DATABASE TODAY

Distributed RDBMS

Apache Hadoop

 **Map Reduce:** (2 stages: first “Map” a job to a node then “Reduce”, where each node process and return

In memory RDBMS

 **Apache SPARK** is both distributed and uses fast in-memory computations

NO-SQL

 Non sql data stores , e.g. Graph storages, Key-value (like “dictionaries” in Python)

Columnar Databases:

Stores columns instead of rows

Useful for data cubes and aggregations

Becoming less popular because of the “in-memory” analytics nowadays

AGENDA FOR TODAY



~~Administration~~



~~Part #1: Database architecture on the web~~



~~Part #2: Databases usage in the last 20K years~~



**Part #3: Introduction to Mysql and
how to do HW#1**

MYSQL: INTRODUCTION

What is MySQL?

 A relational database management system (RDBMS)

 Free and open-source software written in C and C++

Why do we learn MySQL?

 It's the most common database in the web (client-server model)

 Uses by: Facebook, Google, Twitter,

 Is super simple (comparing to Oracle, PostgreSQL)

3 things you (maybe) didn't know about MySQL

 First version was out on 1995

 It is actually owned by Oracle, since 2010

 When it happened, one of the founders quit and forked **Maria-DB** which is still free under the GNU license



DATABASES POPULARITY

Rank			DBMS	Database Model	Score		
Nov 2018	Oct 2018	Nov 2017			Nov 2018	Oct 2018	Nov 2017
1.	1.	1.	Oracle	Relational DBMS	1301.11	-18.16	-58.94
2.	2.	2.	MySQL	Relational DBMS	1159.89	-18.22	-162.14
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1051.55	-6.78	-163.53
4.	4.	4.	PostgreSQL	Relational DBMS	440.24	+20.85	+60.33
5.	5.	5.	MongoDB	Document store	369.48	+6.30	+39.01
6.	6.	6.	DB2	Relational DBMS	179.87	+0.19	-14.19
7.	7.	9.	Redis	Key-value store	144.17	-1.12	+22.99
8.	8.	10.	Elasticsearch	Search engine	143.46	+1.13	+24.05
9.	9.	7.	Microsoft Access	Relational DBMS	138.44	+1.64	+5.12
10.	11.	11.	SQLite	Relational DBMS	122.71	+5.96	+9.95
11.	10.	8.	Cassandra	Wide column store	121.74	-1.64	-2.47
12.	13.	15.	Splunk	Search engine	80.37	+3.48	+15.50
13.	12.	12.	Teradata	Relational DBMS	79.31	+0.67	+1.07
14.	14.	18.	MariaDB	Relational DBMS	73.25	+0.12	+17.96
15.	16.	19.	Hive	Relational DBMS	64.57	+3.47	+11.32

MYSQL: META-DATA

Information_schema

 MySQL server has a default database called “information_schema”

 TABLES table contains information about each table in the database. e.g, name, type, number of rows etc.

 COLUMNS table contains information about each column, such as the table it's belong to, the data type, etc.

 USER_PRIVILEGES table contains information about the users listed in the database (do not confuse with web-users accessing the website).

MYSQL: META-DATA

MySQL Data types

 Each column has a predefined type and possibly a default value

★ Integers: TINYINT, MEDIUMINT, BIGINT

★ Strings: VARCHAR (strings), BLOB (for binaries)

★ Dates: TIMESTAMP, DATE, DATETIME

 Set when the database schema is created

MYSQL: META-DATA

MySQL users privileges

 Root user: granting permissions, creating users, altering creating and deleting data

 Application users: usually read only, no grant.

 Don't ever use root user in a DB connection string (we will discuss it over the next recitations)

OK, SO HOW TO DO HW#1?

 The first thing you have to learn before starting HW#1 is **how to connect to the university MySQL database server**

 Our MySQL server is an internal sever and you will use it both in the final project and in HW#1

 There are several options depending on:

Whether you are **connected to the university network** or not

Your computer's **operating system**.

OK, SO HOW TO DO HW#1?

Connection details:

 Server name: mysqlsrv.cs.tau.ac.il

 DB name: sakila

 user name: sakila

 password: sakila

For security reasons, connection is over SSH

We use SQL “Clients”:

 CLI (command-line interface), mainly for l337 h4x0r\$

 SQL GUI Client:(e.g. Heidi, Sequel-pro, DBeaver, Workbench)

WAIT-A-MINUTE: SSH?

Secure Shell (SSH)

- ★ A network (layer 7) protocol
- ★ Providing secured channel to a remote host.
- ★ Built-in client in Unix based systems
- ★ Putty/Plink is required in Windows based systems.



MYSQL: CLI +SSH



Command line connection (unix)

```
➔ ~ ssh amitsome@nova.cs.tau.ac.il  
amitsome@nova.cs.tau.ac.il's password:  
Last login: Mon Mar 14 22:44:02 2016 from 37.142.245.121  
nova 1% █
```

```
[nova 1% mysql -h mysqlsrv.cs.tau.ac.il -u sakila -p  
[Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 1368667  
Server version: 5.5.35-1ubuntu1-log (Ubuntu)
```

MYSQL: CLI +SSH



Command line connection (unix)

```
[mysql> select 1;
+----+
| 1  |
+----+
1 row in set (0.00 sec)
```

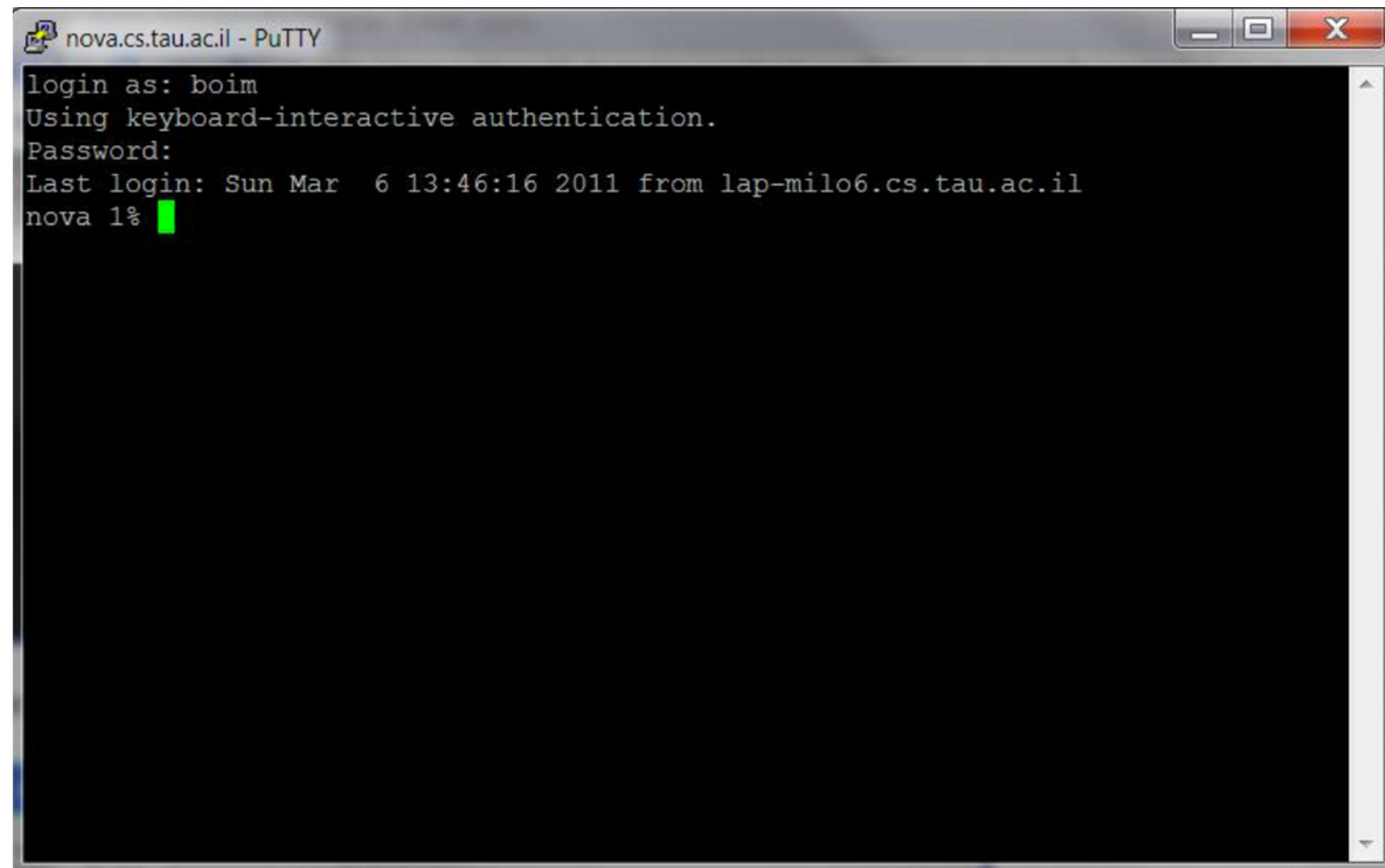
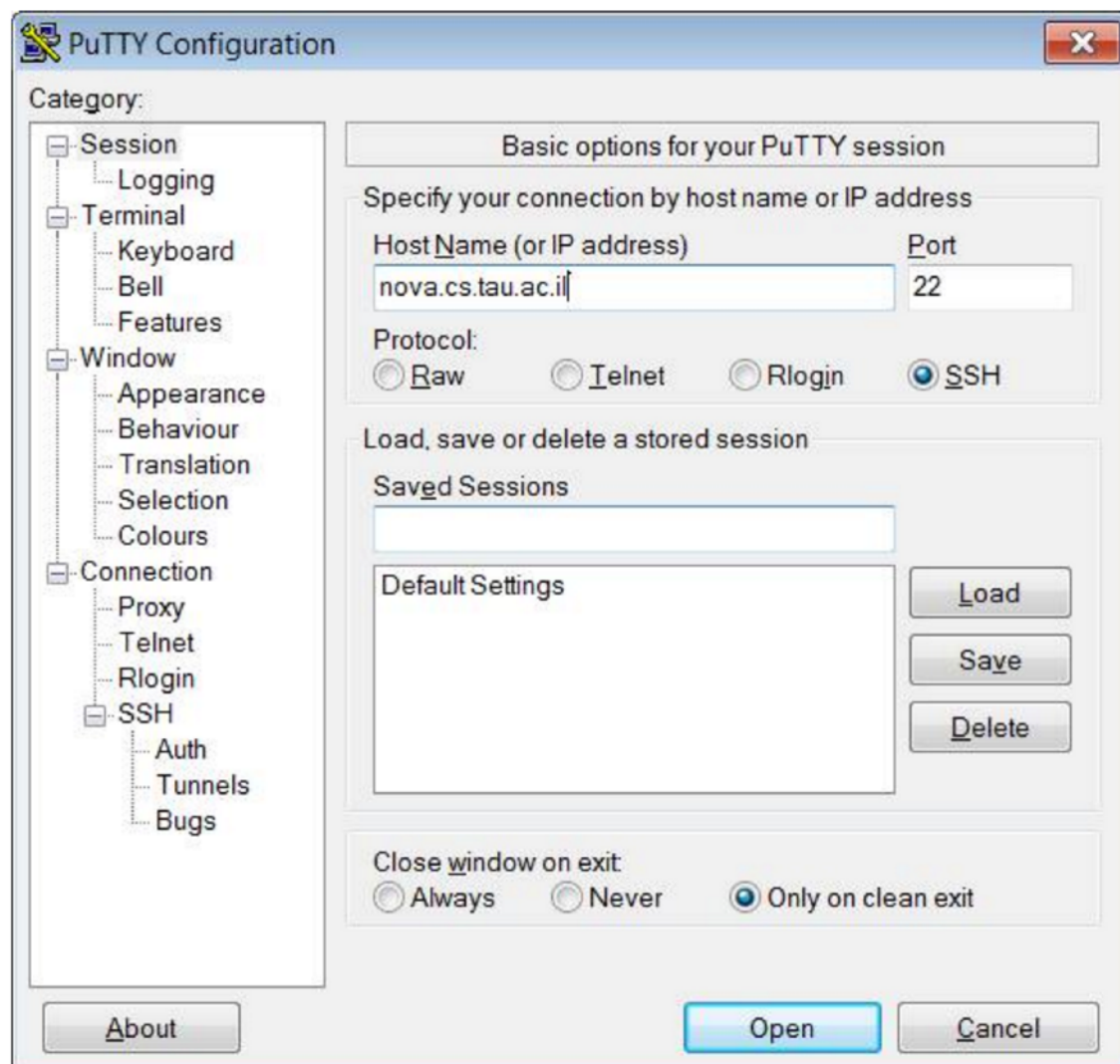


Use Mysql CLI tool to connect to mysqlsrv.tau.ac.il

MYSQL: CLI +SSH

Command line connection (Windows)

Using Putty to Nova



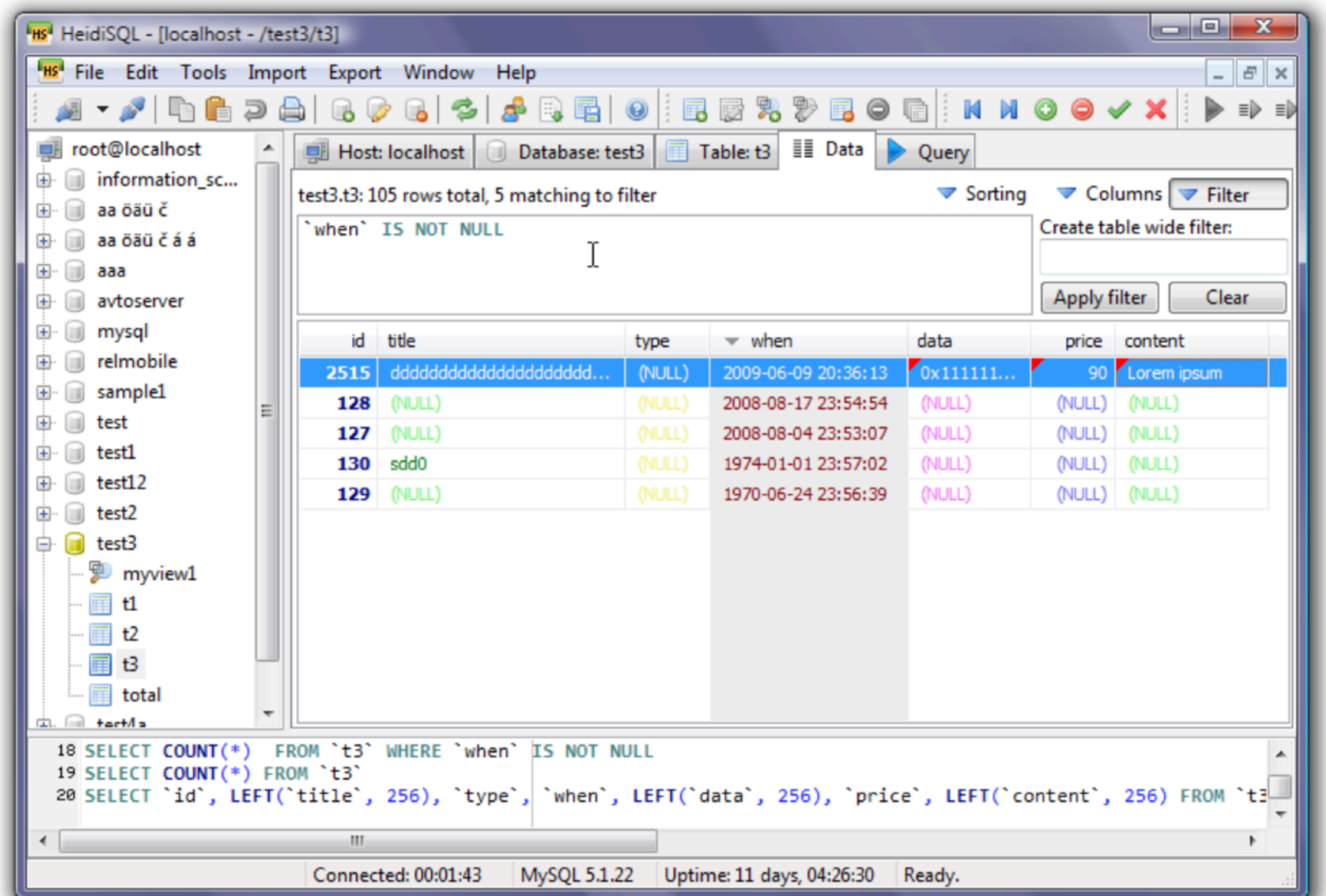
MYSQL: GUI CLIENT+SSH

SQL Client: HeidiSQL

 HeidiSQL is an open source SQL client for Windows

 Mac users: use SequelPRO

 Linux users: use DBeaver



MYSQL: GUI CLIENT+SSH

SQL Software: HeidiSQL

 Things you need:

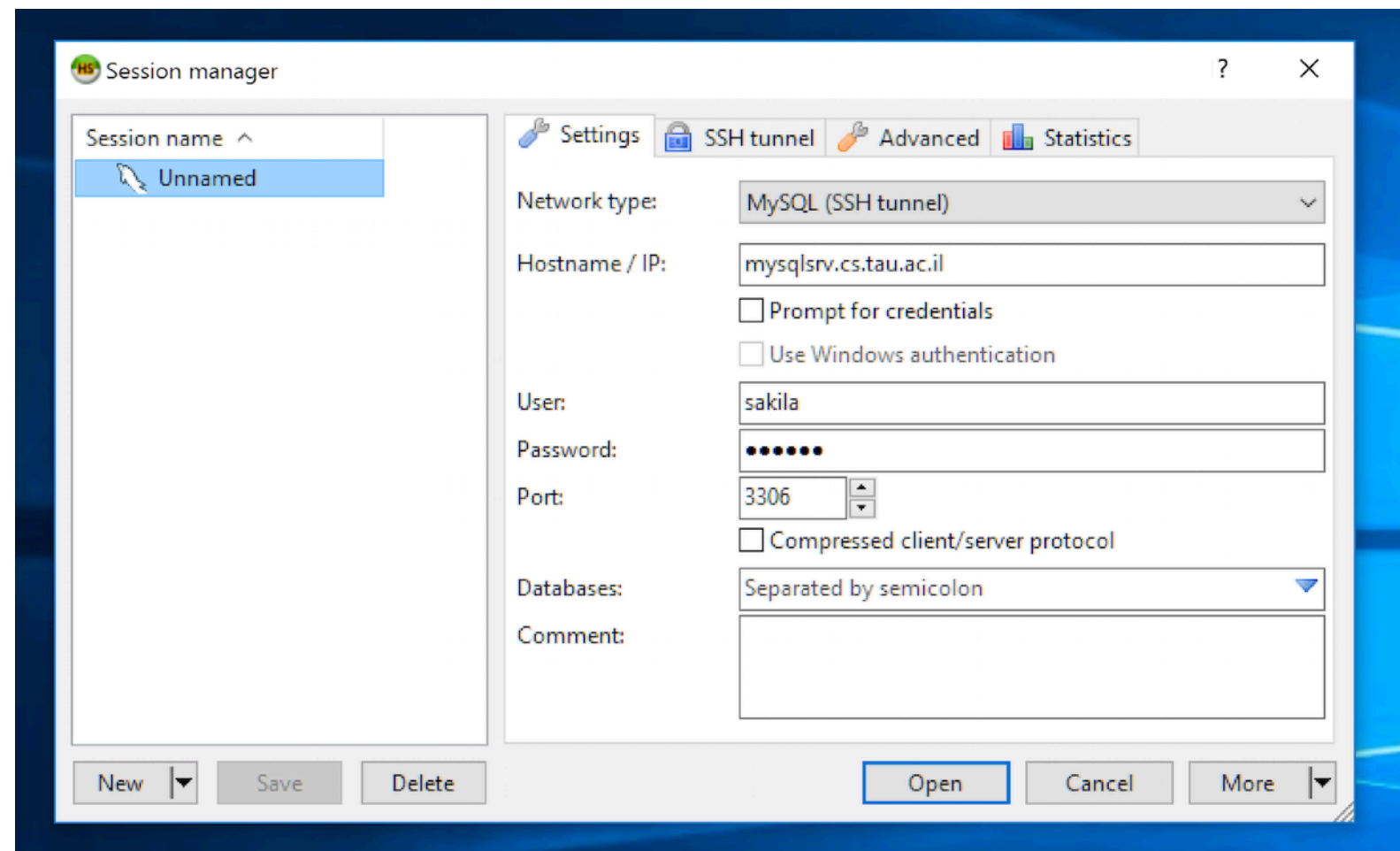
1. HeidiSQL installation
2. Plink.exe (from the Putty bundle)
3. This guide for connecting HeidiSQL via SSH



MYSQL: GUI CLIENT+SSH

How to connect to the HW#1 database:

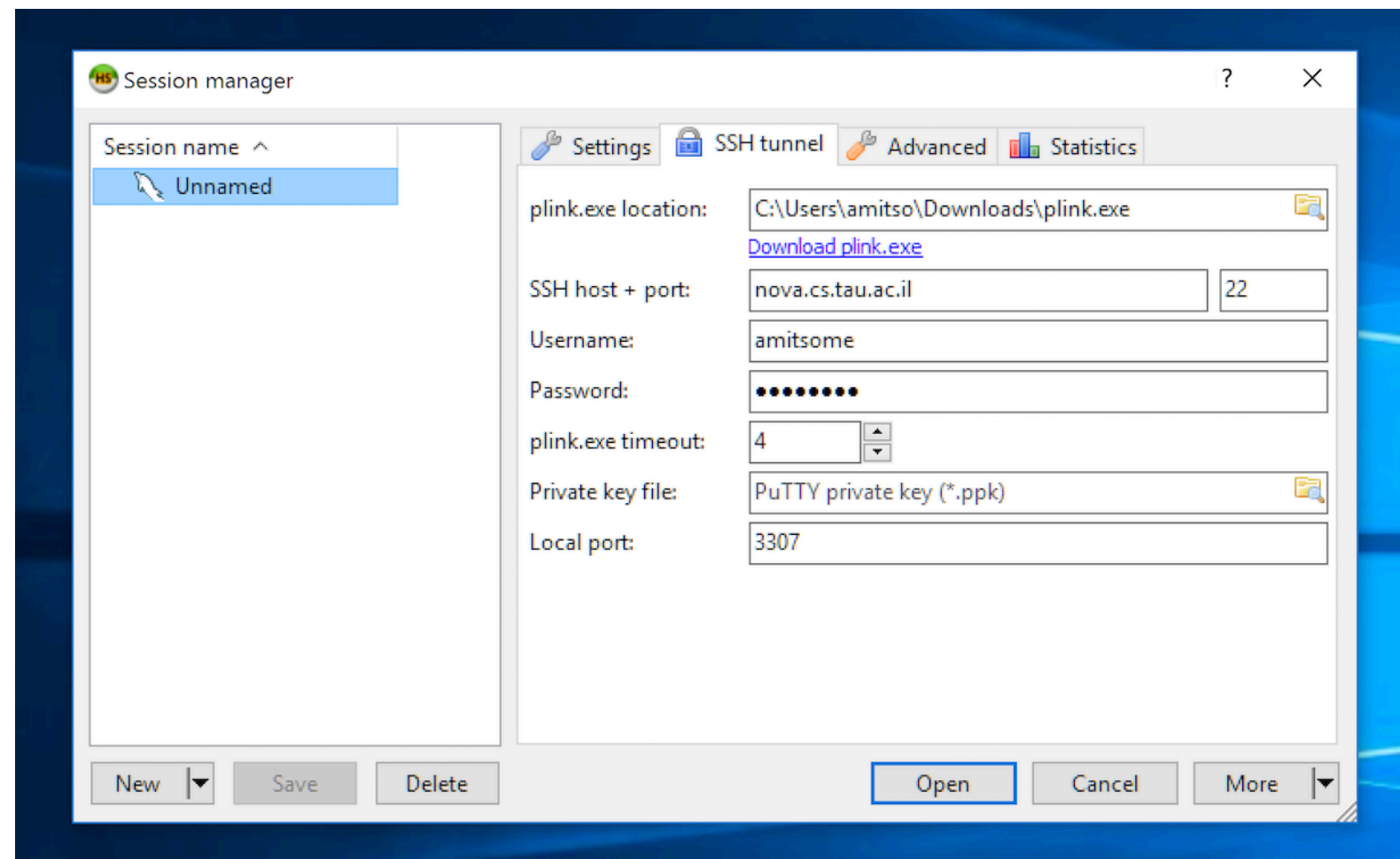
1. Open HeidiSQL
2. Enter the MySQL server details: mysqlsrv.cs.tau.ac.il
3. Enter the user&password: sakila (for both)



MYSQL: GUI CLIENT+SSH

How to connect to the HW#1 database:

4. Go to the SSH tunnel tab
5. Enter the path to plink.exe
6. Insert the address to Nova and your university user and password
7. Click “Open”

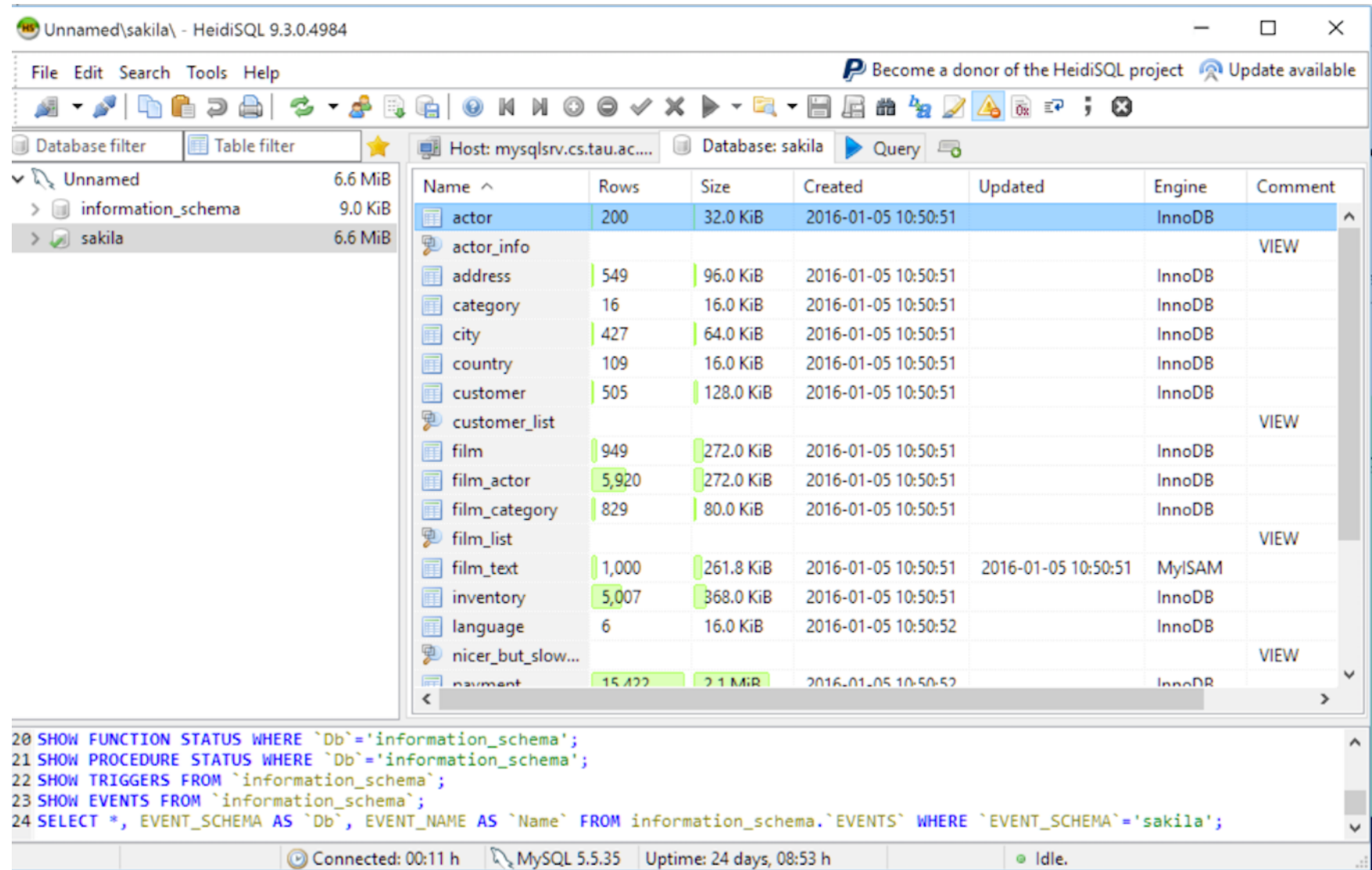


MYSQL: GUI CLIENT+SSH

 **How to connect to the HW#1 database:**

8. Choose the “sakila” database

9. Write queries



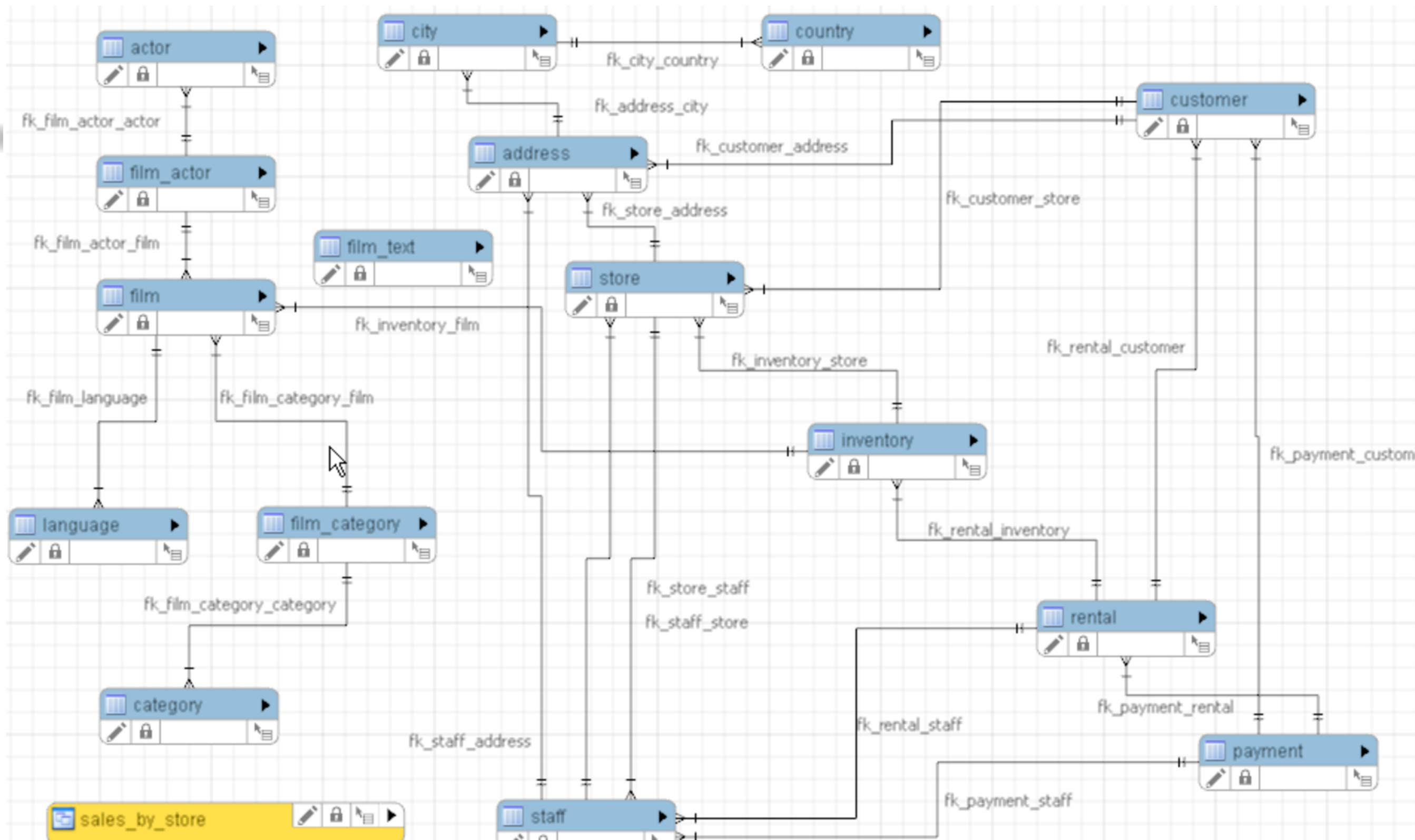
The screenshot shows the HeidiSQL 9.3.0.4984 interface. The left sidebar displays the database structure with 'sakila' selected. The main pane shows a table list for the 'sakila' database. The bottom pane contains a SQL query window with the following queries:

```
20 SHOW FUNCTION STATUS WHERE `Db`='information_schema';
21 SHOW PROCEDURE STATUS WHERE `Db`='information_schema';
22 SHOW TRIGGERS FROM `information_schema`;
23 SHOW EVENTS FROM `information_schema`;
24 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='sakila';
```

The status bar at the bottom indicates: Connected: 00:11 h, MySQL 5.5.35, Uptime: 24 days, 08:53 h, Idle.

Name	Rows	Size	Created	Updated	Engine	Comment
actor	200	32.0 KiB	2016-01-05 10:50:51		InnoDB	
actor_info						VIEW
address	549	96.0 KiB	2016-01-05 10:50:51		InnoDB	
category	16	16.0 KiB	2016-01-05 10:50:51		InnoDB	
city	427	64.0 KiB	2016-01-05 10:50:51		InnoDB	
country	109	16.0 KiB	2016-01-05 10:50:51		InnoDB	
customer	505	128.0 KiB	2016-01-05 10:50:51		InnoDB	
customer_list						VIEW
film	949	272.0 KiB	2016-01-05 10:50:51		InnoDB	
film_actor	5,920	272.0 KiB	2016-01-05 10:50:51		InnoDB	
film_category	829	80.0 KiB	2016-01-05 10:50:51		InnoDB	
film_list						VIEW
film_text	1,000	261.8 KiB	2016-01-05 10:50:51	2016-01-05 10:50:51	MyISAM	
inventory	5,007	368.0 KiB	2016-01-05 10:50:51		InnoDB	
language	6	16.0 KiB	2016-01-05 10:50:52		InnoDB	
nicer_but_slow...						VIEW
payment	15,432	2.1 MiB	2016-01-05 10:50:52		InnoDB	

MYSQL: SAKILA SCHEMA



MYSQL: SAKILA SCHEMA



Example Query:

```
mysql> SELECT CONCAT(customer.last_name, ' ', customer.first_name) AS customer,  
-> address.phone, film.title  
-> FROM rental INNER JOIN customer ON rental.customer_id = customer.customer_id  
-> INNER JOIN address ON customer.address_id = address.address_id  
-> INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id  
-> INNER JOIN film ON inventory.film_id = film.film_id  
-> WHERE rental.return_date IS NULL  
-> AND rental_date + INTERVAL film.rental_duration DAY < CURRENT_DATE()  
-> LIMIT 5;
```


MYSQL: SAKILA SCHEMA



Example Query:

```
mysql> SELECT CONCAT(customer.last_name, ' ', customer.first_name) AS customer,  
-> address.phone, film.title  
-> FROM rental INNER JOIN customer ON rental.customer_id = customer.customer_id  
-> INNER JOIN address ON customer.address_id = address.address_id  
-> INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id  
-> INNER JOIN film ON inventory.film_id = film.film_id  
-> WHERE rental.return_date IS NULL  
-> AND rental_date + INTERVAL film.rental_duration DAY < CURRENT_DATE()  
-> LIMIT 5;
```



Results:

customer	phone	title
OLVERA, DWAYNE	62127829280	ACADEMY DINOSAUR
HUEY, BRANDON	99883471275	ACE GOLDFINGER
BROWN, ELIZABETH	10655648674	AFFAIR PREJUDICE
OWENS, CARMEN	272234298332	AFFAIR PREJUDICE
HANNON, SETH	864392582257	AFRICAN EGG

YOUR BEST FRIENDS

 MySQL is the most common database used on the web.

 Therefore, **stackoverflow** is your friend.

 Another good friend you got : w3schools.com. for everything you need regarding web development and basic SQL use.

 MySQL cheatsheet:

 <https://en.wikibooks.org/wiki/MySQL/CheatSheet>