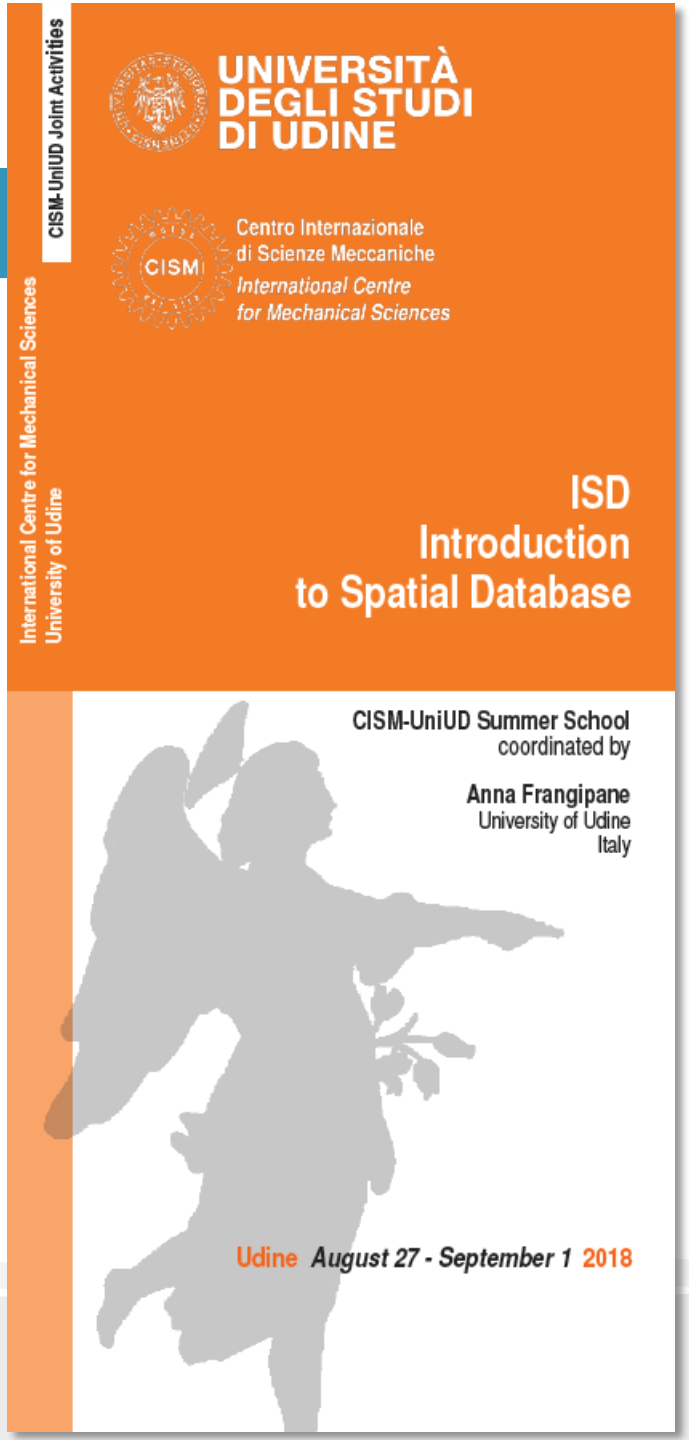


Introduction to (My)SQL

Marco BASALDELLA, PhD

Department of Mathematics, Computer Science and Physics
University of Udine
marco.basaldella@uniud.it | www.basaldella.it

Outline

1. Database Management Systems
2. The Structured Query Language
3. Retrieving Data from a Database

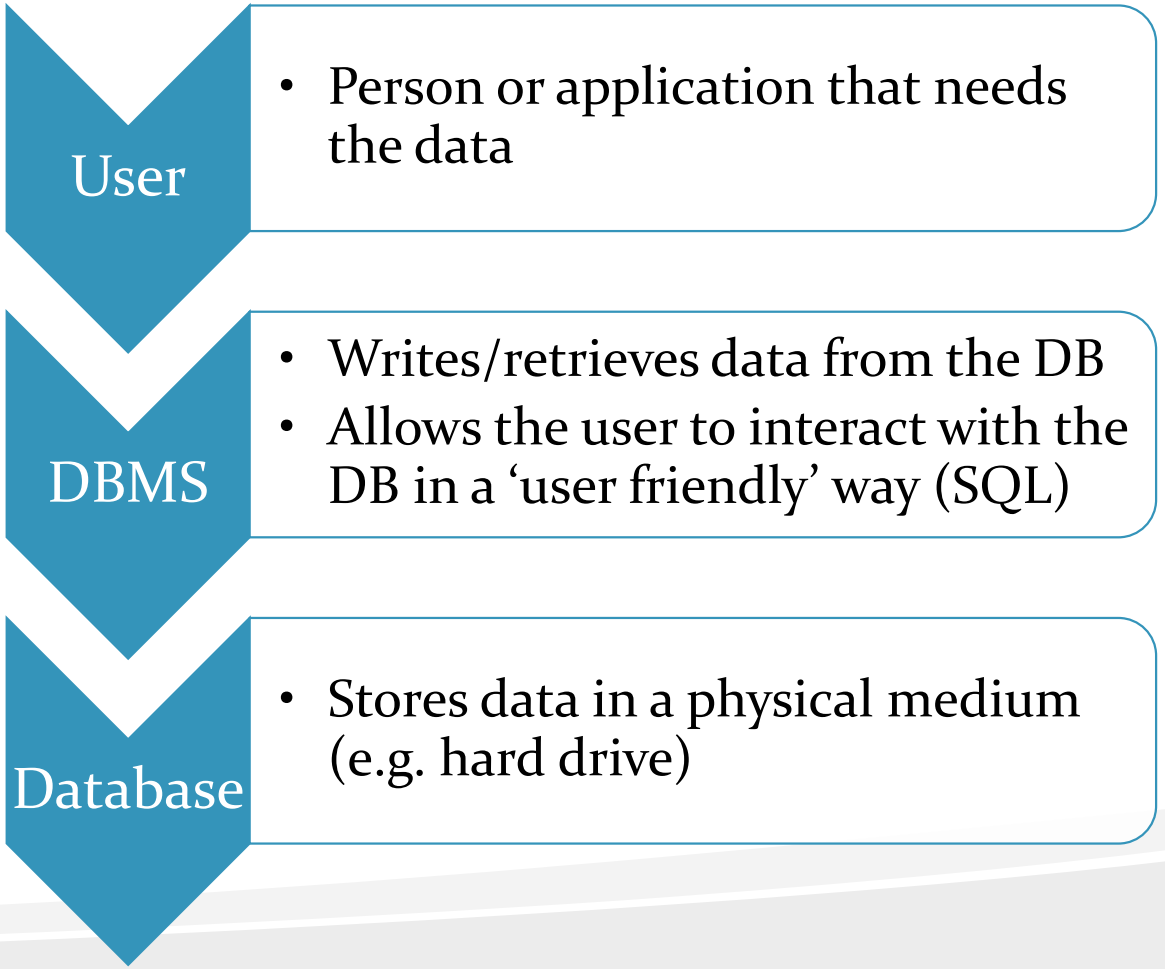
All the material is available online:

`https://github.com/basaldella/isd2018`

Database Management Systems

Database Management Systems

- **Database**: collection of data, tables, …
- **DBMS**: software that allows the interaction and management of databases



Database Management Systems/2

Duties of a DMBS:

- Store the database **schema, data, tables**… efficiently
  - Data may be stored in a single file, or in dozens of computers
- Guarantee the **integrity** of the data
  - e.g.: prevent concurrency: users that write on the same table at the same moments
- Guarantee the **security** of the data
  - e.g.: prevent unauthorized operations: users that try to access data owned by other users

Database Management Systems/3

Popular **relational** DBMSs:

- Oracle RDBMS (commercial)
- Microsoft SQL Server (commercial)
- PostgreSQL (Open Source Software)
- **MySQL** (OSS)

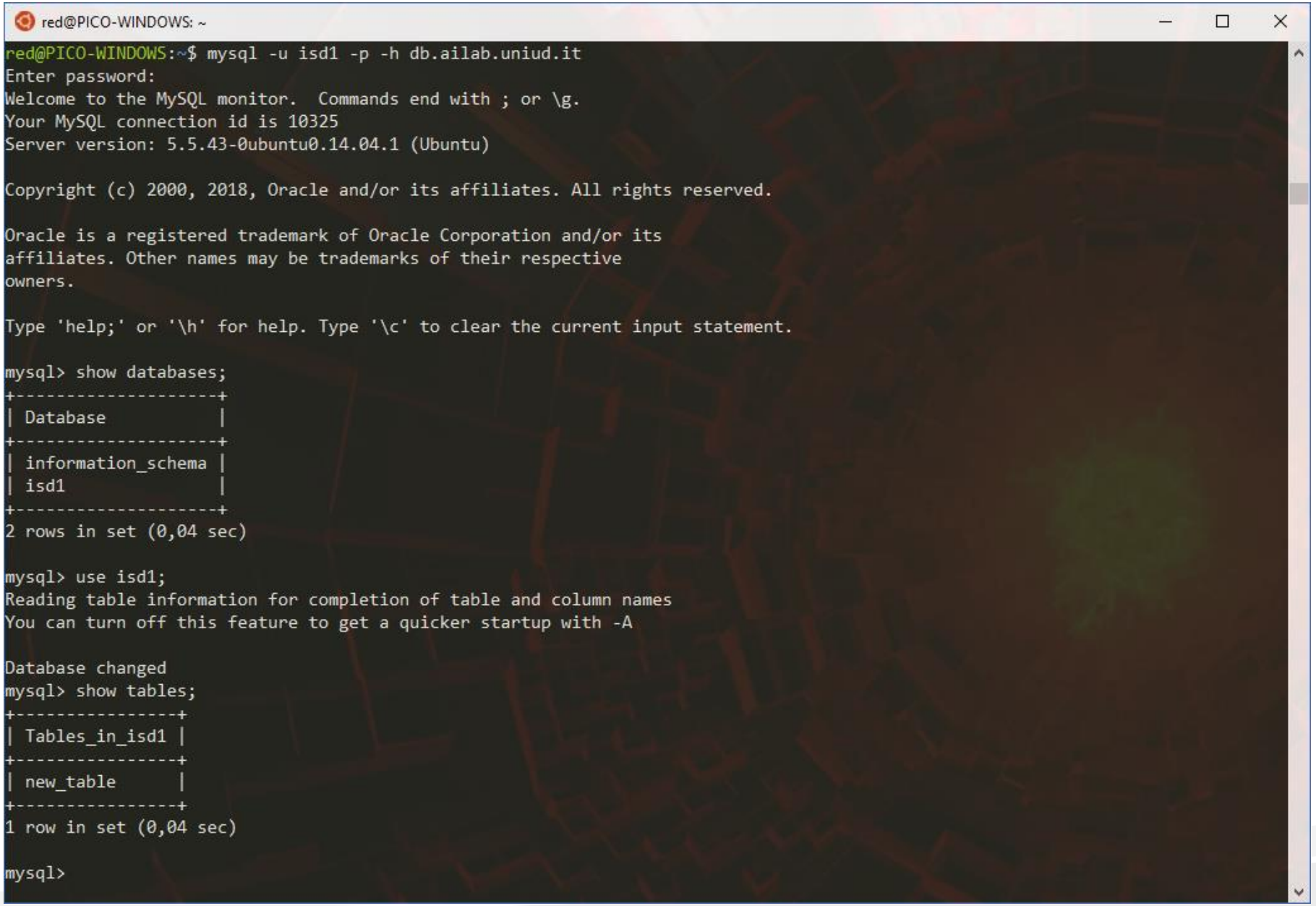
Other non-relational DBMSs: MongoDB, Redis, Zope…

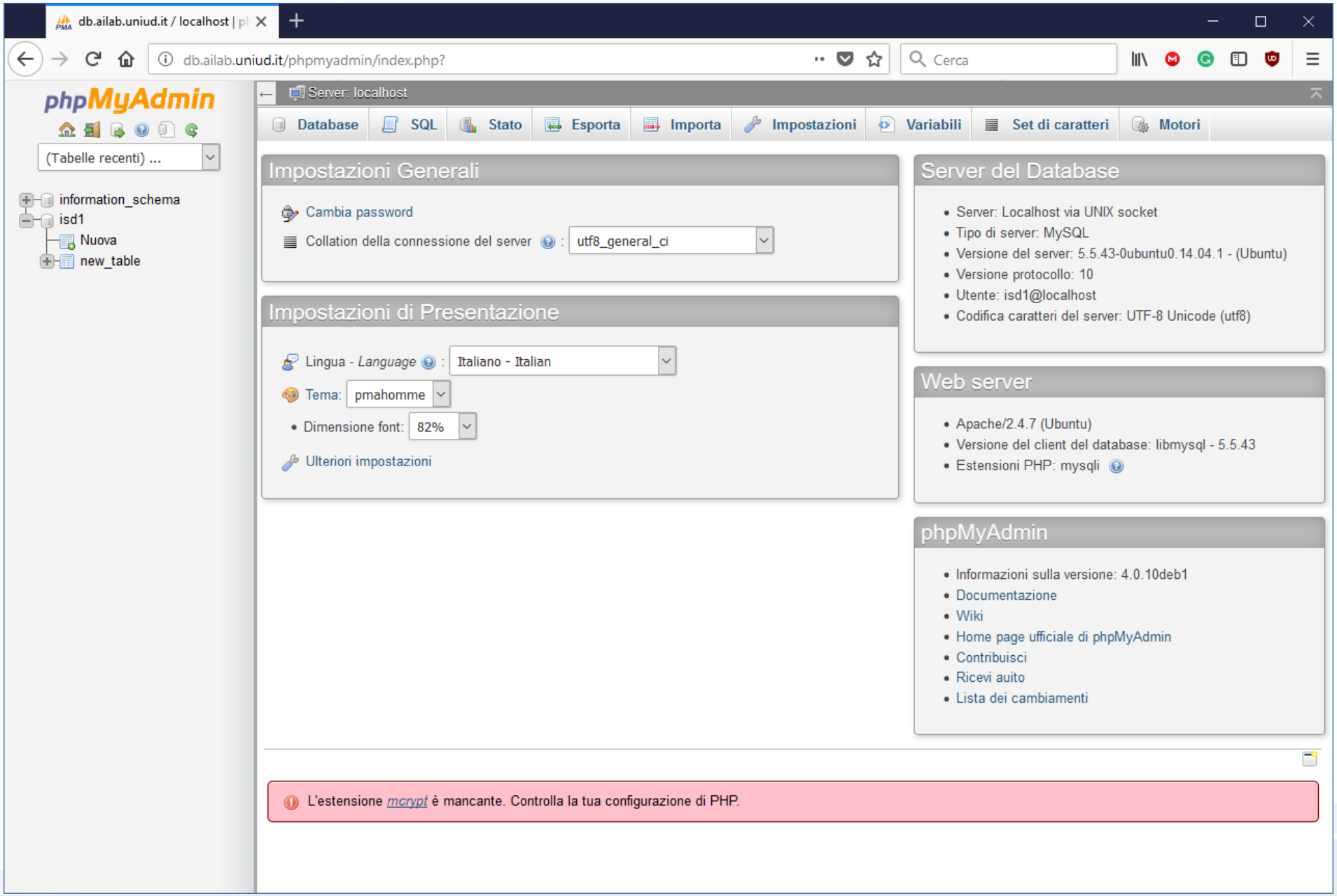
- Free, open source relational database management system
- Paid version for enterprise customers
- “my”: name of the daughter of the founder
- Logo: Sakila the dolphin
- Founded in 1995, 2008 acquired by Sun Microsystems, 2010 acquired by Oracle
- MySQL users include **Facebook, Twitter, airbnb, Amazon, booking.com, NASA, Wordpress, Lufthansa…**

MySQL/2

MySQL is composed by two parts:

- A **server**, which stores the data, and
- A **client**, which allows users and applications to run **SQL commands** on the server
- (human) Users can interact with the server with the **command-line client**, via **web clients**, or with specialized softwares


red@PICO-WINDOWS: ~
red@PICO-WINDOWS:~$ mysql -u isd1 -p -h db.ailab.uniud.it
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10325
Server version: 5.5.43-0ubuntu0.14.04.1 (Ubuntu)
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> show databases;
| Database |
| information_schema |
| isd1 |
2 rows in set (0,04 sec)
mysql> use isd1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> show tables;
| Tables_in_isd1 |
| new_table |
1 row in set (0,04 sec)
mysql>


db.ailab.uniud.it / localhost | p
db.ailab.uniud.it/phpmyadmin/index.php?
Cerca
phpMyAdmin
(Tabelle recenti) ...
information_schema
isd1
Nuova
new_table
Server: localhost
Database
SQL
Stato
Esporta
Importa
Impostazioni
Variabili
Set di caratteri
Motori
Impostazioni Generali
Cambia password
Collation della connessione del server
utf8_general_ci
Impostazioni di Presentazione
Lingua - Language
Italiano - Italian
Tema:
pmahomme
Dimensione font:
82%
Ulteriori impostazioni
Server del Database
Server: Localhost via UNIX socket
Tipo di server: MySQL
Versione del server: 5.5.43-0ubuntu0.14.04.1 - (Ubuntu)
Versione protocollo: 10
Utente: isd1@localhost
Codifica caratteri del server: UTF-8 Unicode (utf8)
Web server
Apache/2.4.7 (Ubuntu)
Versione del client del database: libmysql - 5.5.43
Estensioni PHP: mysqli
phpMyAdmin
Informazioni sulla versione: 4.0.10deb1
Documentazione
Wiki
Home page ufficiale di phpMyAdmin
Contribuisci
Ricevi auito
Lista dei cambiamenti
L'estensione mcrypt è mancante. Controlla la tua configurazione di PHP.

Structured Query Language

SQL

- Invented in the 1970s by IBM
- Standardized by ISO in 1987 (SQL-87)
  - Updated about every three years
  - Last standard: SQL:2016

- Each DBMS offers a *dialect* of SQL which works *roughly* the same
  - In practice, code is often **not** portable between different DBMSs

SQL/2

We can see SQL as a family of *computer languages*:

- Data query language (DQL),
- Data definition language (DDL)
- Data control language (DCL)
- Data manipulation language (DML).

Note: different scholars group the languages differently, e.g. sometimes DQL is merged in DML.

Data Control Language

Used to

- Create users, change passwords
  - `CREATE USER 'newuser' IDENTIFIED BY 'password';`
- Manage the privileges of the users in the database
  - e.g. allow user `john_smith` to access the table `foo`
  - e.g. prevent user `jane_doe` to modify the table `bar`
- We **won't** use DCL

Data Query Language

Data Query Language

Used to **query** the database, i.e. to **retrieve the data**

Example:

- retrieve the table ‘foo’:
  ```
  SELECT * FROM foo
  ```
- retrieve the user with username ‘bar’ from the table ‘users’:
  ```
  SELECT * FROM users WHERE username = ‘bar’
  ```

Data Query Language

Used to **query** the database, i.e. to **retrieve the data**

Examples:

- retrieve the table ‘foo’:

```
SELECT *
FROM foo
```

- retrieve the user with username ‘bar’ from the table ‘users’:

```
SELECT *
FROM users
WHERE username = ‘bar’
```

Data Query Language: example

```
SELECT *
FROM users
WHERE username = ‘bar’
```

WHERE restricts the selection to the rows that match the provided condition

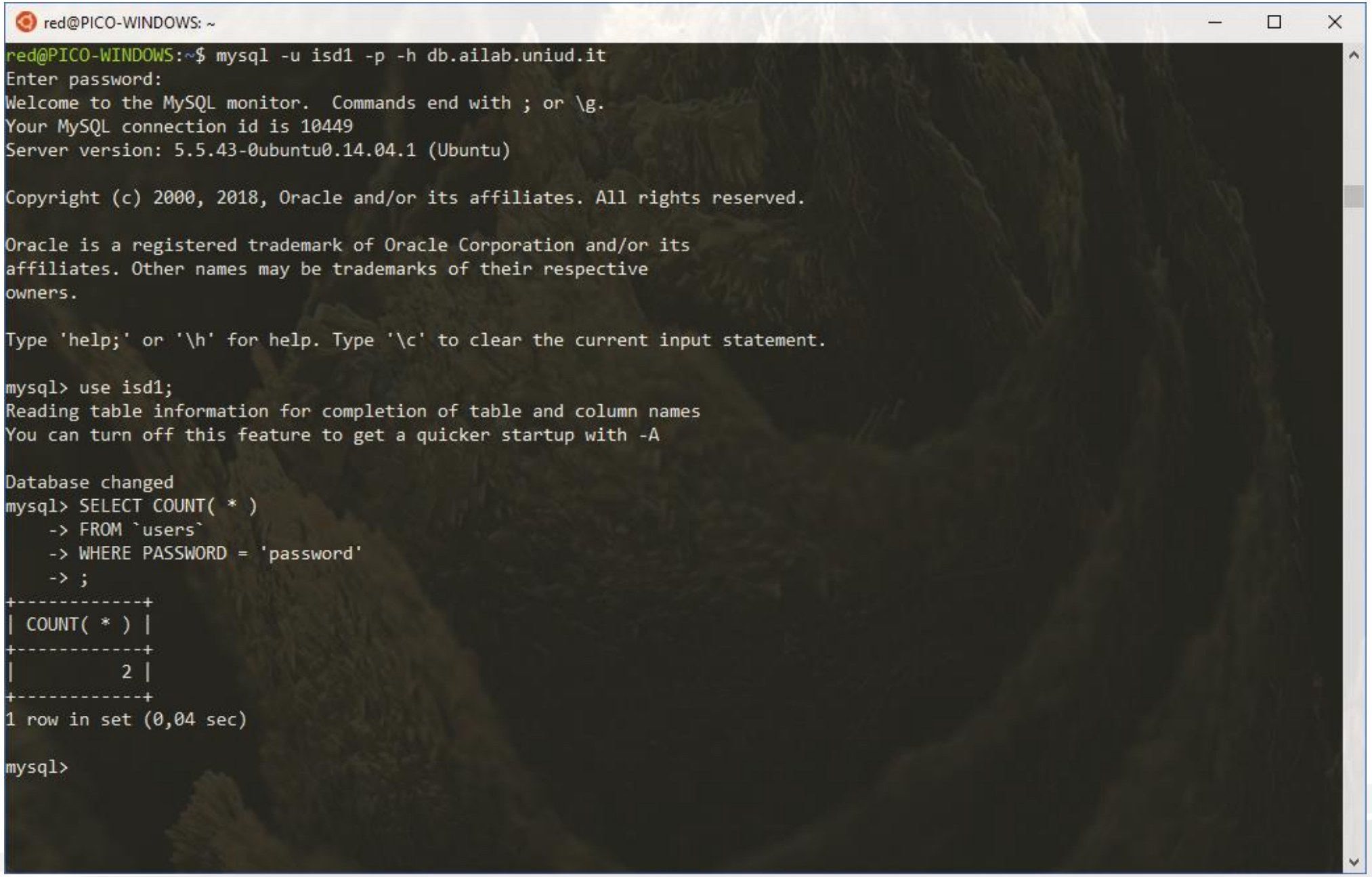
ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password
…	…	…
…	…	…

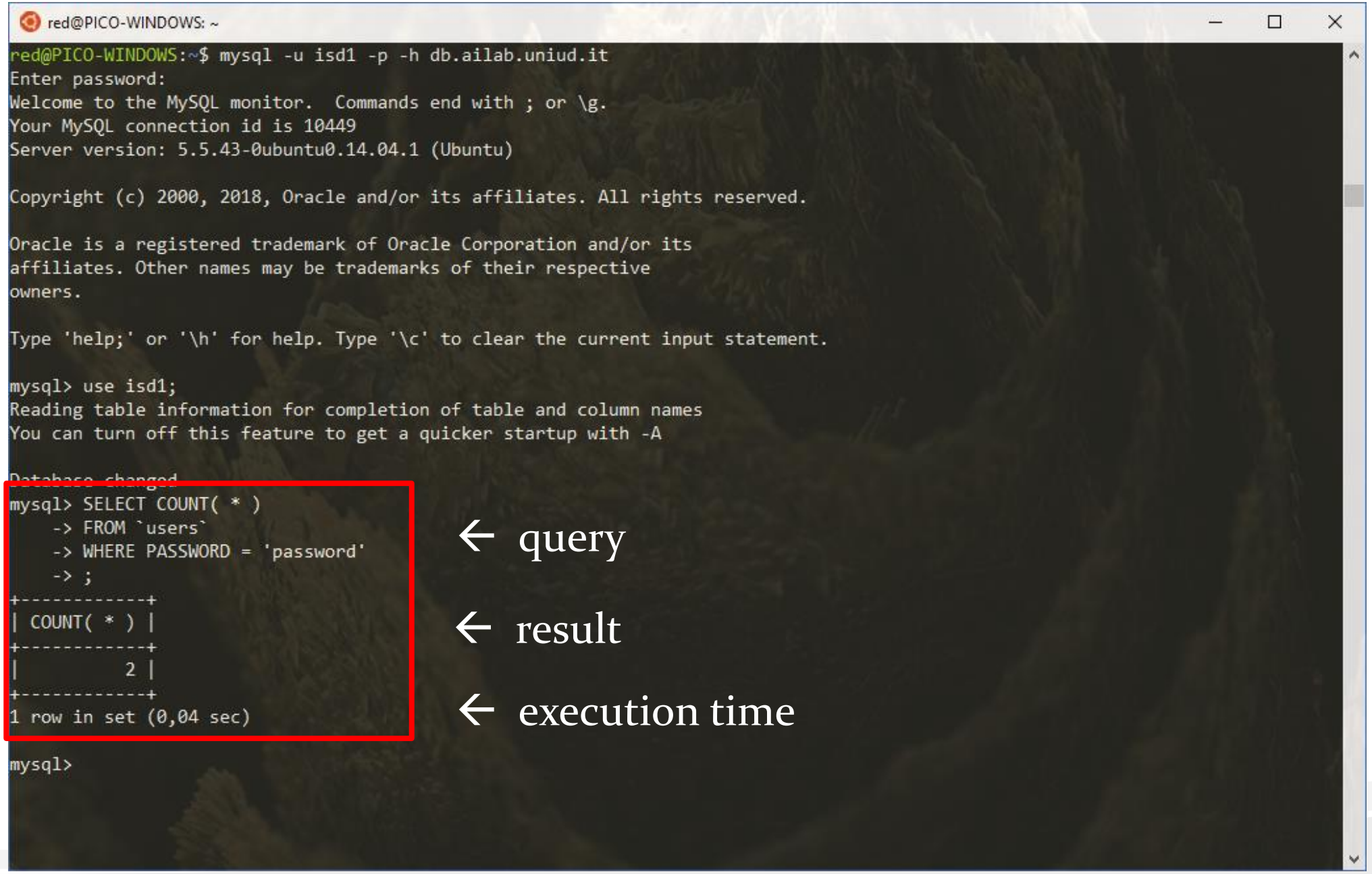
Data Query Language: example/2

```
SELECT COUNT(*)
FROM users
WHERE password =
‘password’
```

COUNT counts the rows that match the condition provided by the WHERE clause

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password
…	…	…
…	…	…


red@PICO-WINDOWS: ~
red@PICO-WINDOWS:~$ mysql -u isd1 -p -h db.ailab.uniud.it
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10449
Server version: 5.5.43-0ubuntu0.14.04.1 (Ubuntu)
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use isd1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> SELECT COUNT(*)
    -> FROM `users`
    -> WHERE PASSWORD = 'password'
    -> ;
+------------+
| COUNT(*) |
+------------+
| 2 |
+------------+
1 row in set (0,04 sec)
mysql>


red@PICO-WINDOWS: ~
red@PICO-WINDOWS:~$ mysql -u isd1 -p -h db.ailab.uniud.it
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10449
Server version: 5.5.43-0ubuntu0.14.04.1 (Ubuntu)
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use isd1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> SELECT COUNT(*)
-> FROM `users`
-> WHERE PASSWORD = 'password'
-> ;
+------------+
| COUNT(*) |
+------------+
| 2 |
+------------+
1 row in set (0,04 sec)
mysql>
← query
← result
← execution time

Data Query Language/concl.

`SELECT` can be combined with **lots** of commands:

- `MIN`, `MAX`, `SUM`, return the minimum, maximum, or the sum of the values for a given column
- `ORDER_BY` orders the results based on one or more attributes
- `GROUP_BY` groups the results based on one or more attributes
- `DISTINCT` returns only different values

Examples

WHERE

```
SELECT *
FROM Students
WHERE Password =
'password'
```

Returns

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

Operators

Operator	Meaning	Example
=	Equals	name = 'john'
!=	Not equals	name != 'john'
>	Greater than	age > 17
>=	Greater or equal than	age >= 18
<	Less than	price < 10
<=	Less or equal than	Price <= 100
AND, OR, NOT	Logical and, or, not	price > 1 AND price < 10

Examples

Multiple operators

```
SELECT *
FROM Students
WHERE Password =
'password‘ AND Id > 3
```

Returns

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

Examples

MIN, MAX, SUM, AVG

```
SELECT MAX(ID)
FROM Students
```

Returns

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

Examples

ORDER BY

```
SELECT *
FROM Students
ORDER BY username
```

Reverse order:

```
ORDER BY field ASC
```

Returns

ID	username	Password
5	anne.other	password
2	bar	password
1	foo	12345
4	jane.doe	isd2018
3	john.smith	99999

GROUP BY

How it works

- Combined with aggregate function (min, max …)
- **Groups** the result set by one or more columns
- **Example:**
  How many people by Country?

Example table

ID	Name	Country
1	Mario Rossi	Italy
2	Max Mustermann	Germany
3	Rosa Bianchi	Italy
4	Giuseppe Verdi	Italy
5	John Smith	U.K.
6	Pierre Toulemonde	France
7	Anne Other	U.K.
8	Jane Doe	U.K.

GROUP BY

How it works

```
SELECT Country, Count(ID)
FROM Students
GROUP BY Country
```

Returns

Country	COUNT(ID)
Italy	3
Germany	1
U.K.	3
France	1

GROUP BY

How it works

```
SELECT Country, Count(ID)
FROM Students
GROUP BY Country
ORDER BY Count(ID) ASC
```

Returns

Country	COUNT(ID)
France	1
Germany	1
U.K.	3
Italy	3

Examples

DISTINCT

```
SELECT
DISTINCT(Password)
FROM Students
```

Returns

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

Nested Queries

Suppose we want to find the users who are banned.

Table **`Banned_Users`**:

ID
2
5

Table **`Users`**:

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

Nested Queries

Suppose we want to find the users who are banned.

Instruction **IN:**

```
SELECT *
FROM Users
WHERE ID IN (2,5)
```

Table **Banned_Users**:

ID
2
5

Table **Users**:

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

Nested Queries

Even better: we can "nest" queries

```
SELECT *
FROM Users
WHERE ID IN (
        SELECT ID
        FROM Banned_Users
);
```

Table **`Banned_Users`**:

ID
2
5

Table **`Users`**:

ID	username	Password
1	foo	12345
2	bar	password
3	john.smith	99999
4	jane.doe	isd2018
5	anne.other	password

MySQL Workbench

Hands-on tutorial

MySQL Workbench

MySQL Workbench is a software that offers a graphical user interface (GUI) to easily manage of MySQL servers. It offers

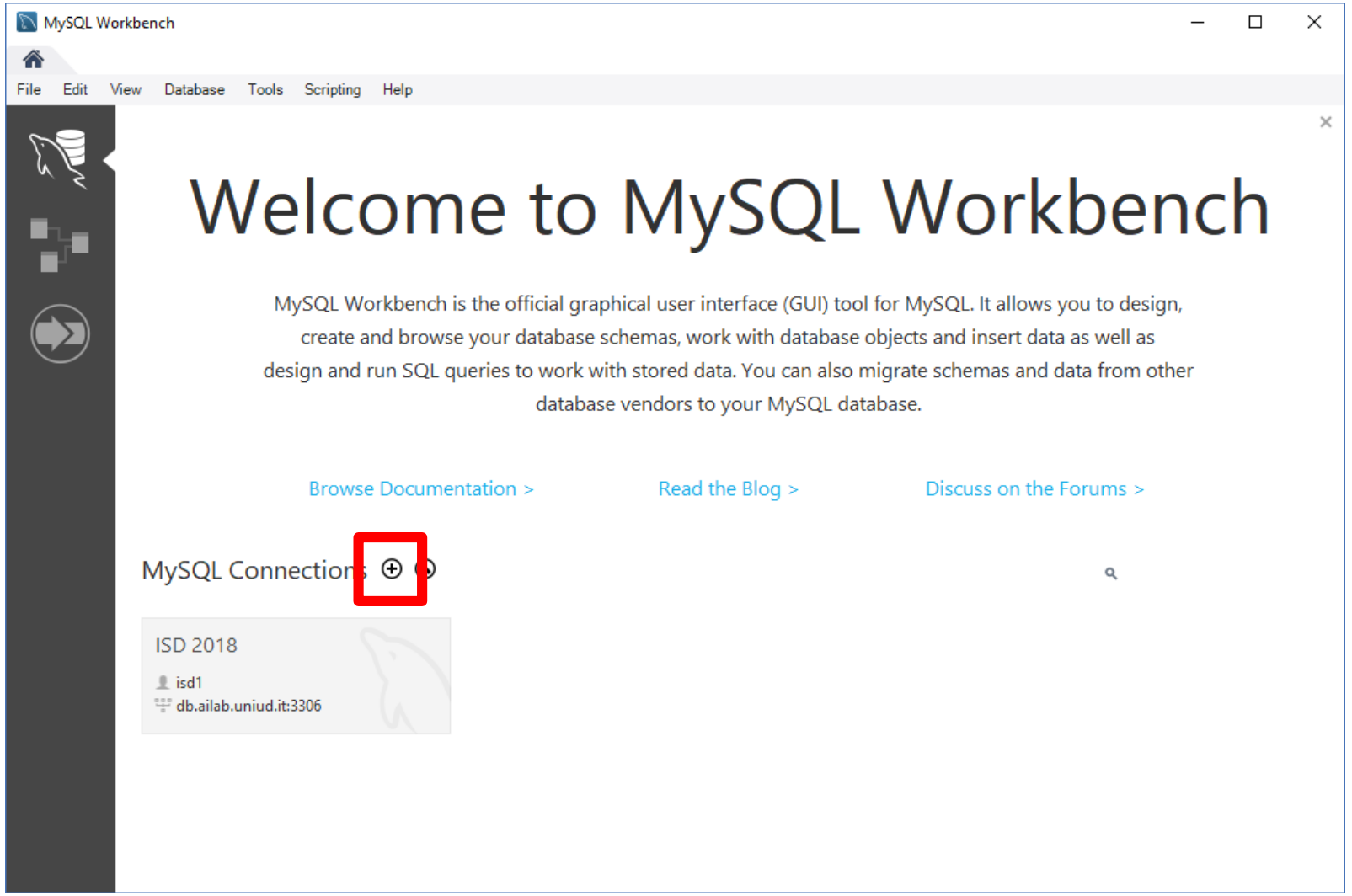
- Database diagrams (EER)
- Database visualization
- SQL console
- Performance monitoring
- …

MySQL Workbench/2

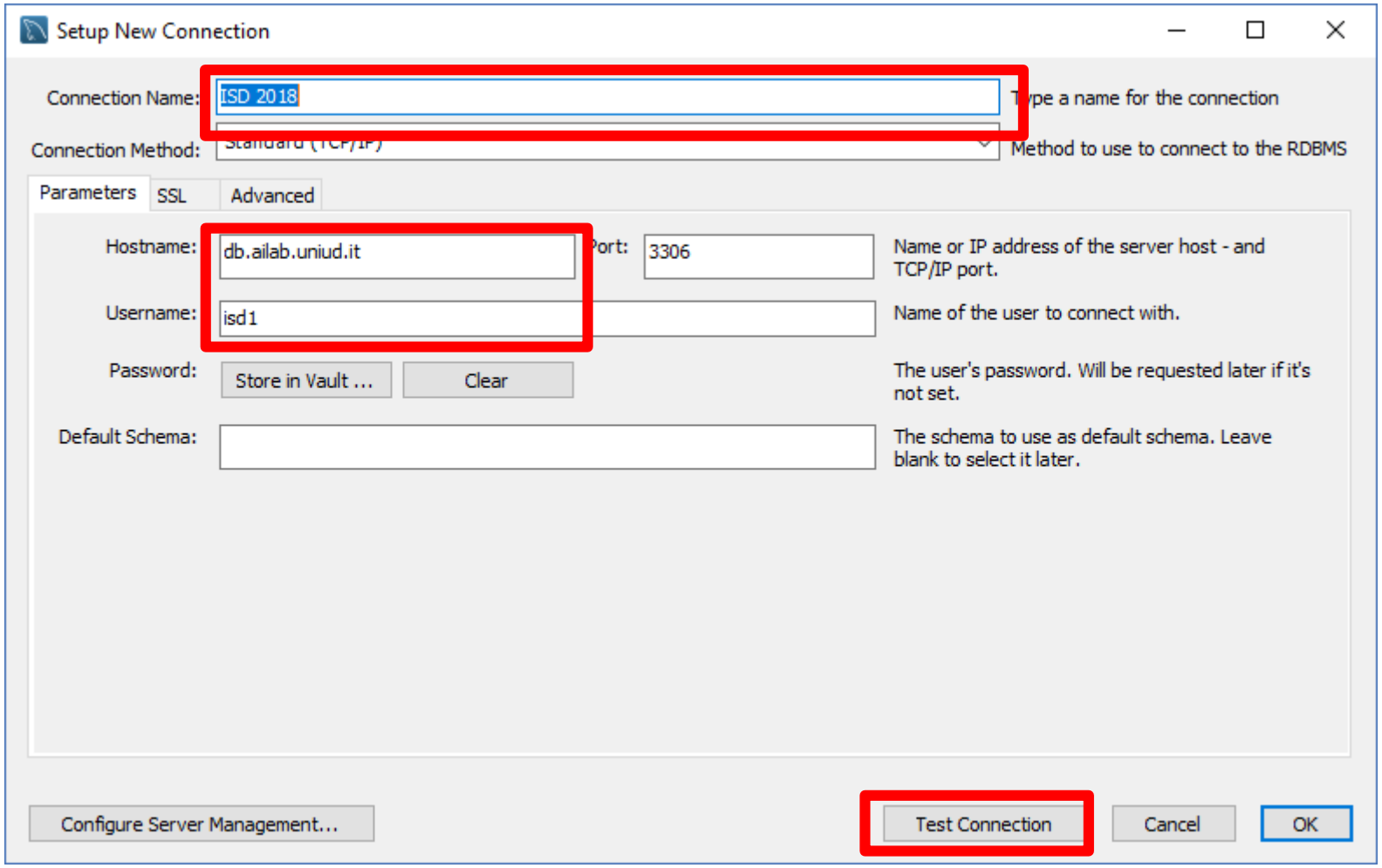
We will use MySQL Workbench as our tool of choice for working with MySQL.

You will work with a database installed in the Artificial Intelligence Laboratory servers. The credentials are:

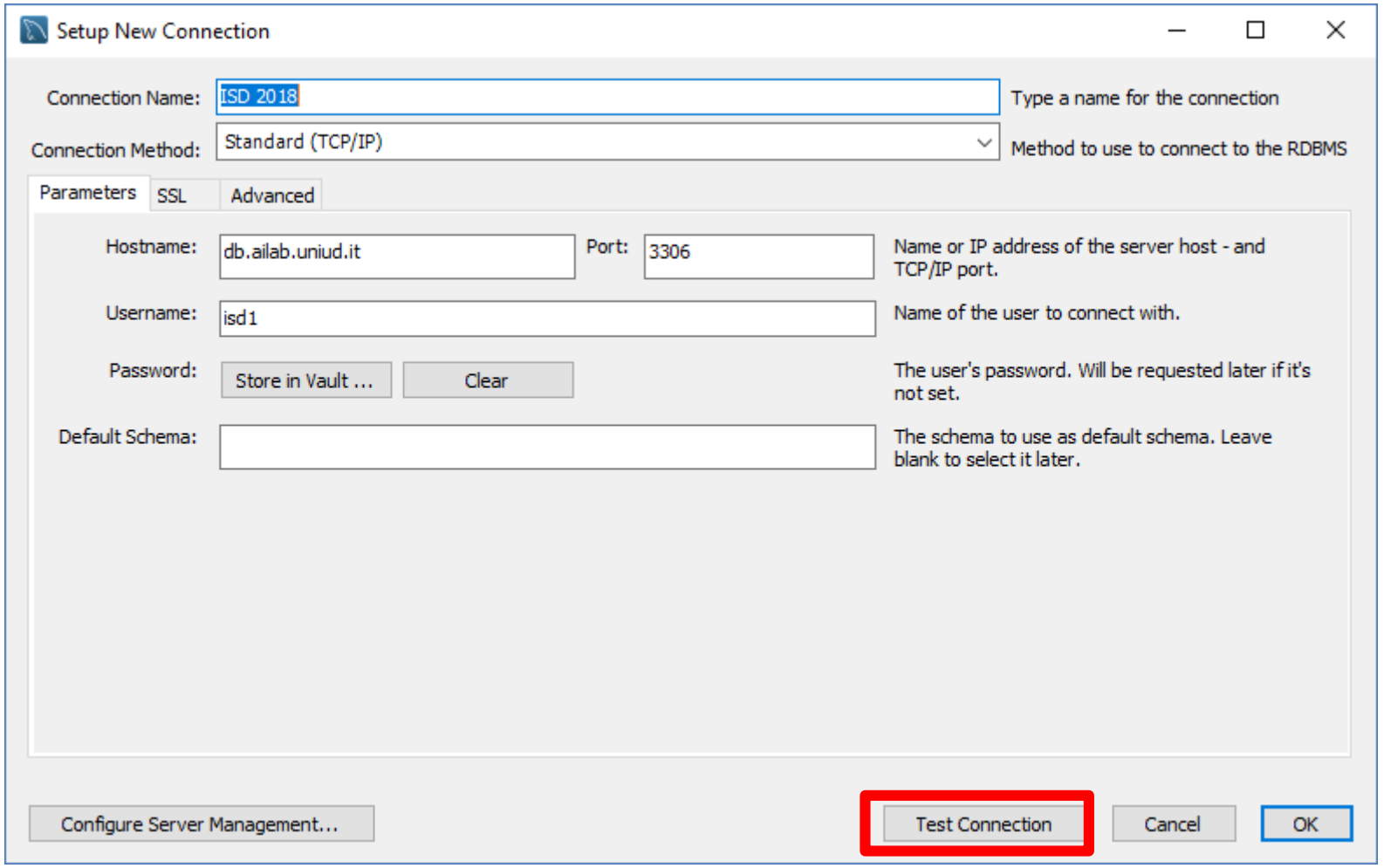
- hostname: `db.ailab.uniud.it`
- username: `isdXX`, where XX is the number of your PC
- password: `ISD2018`



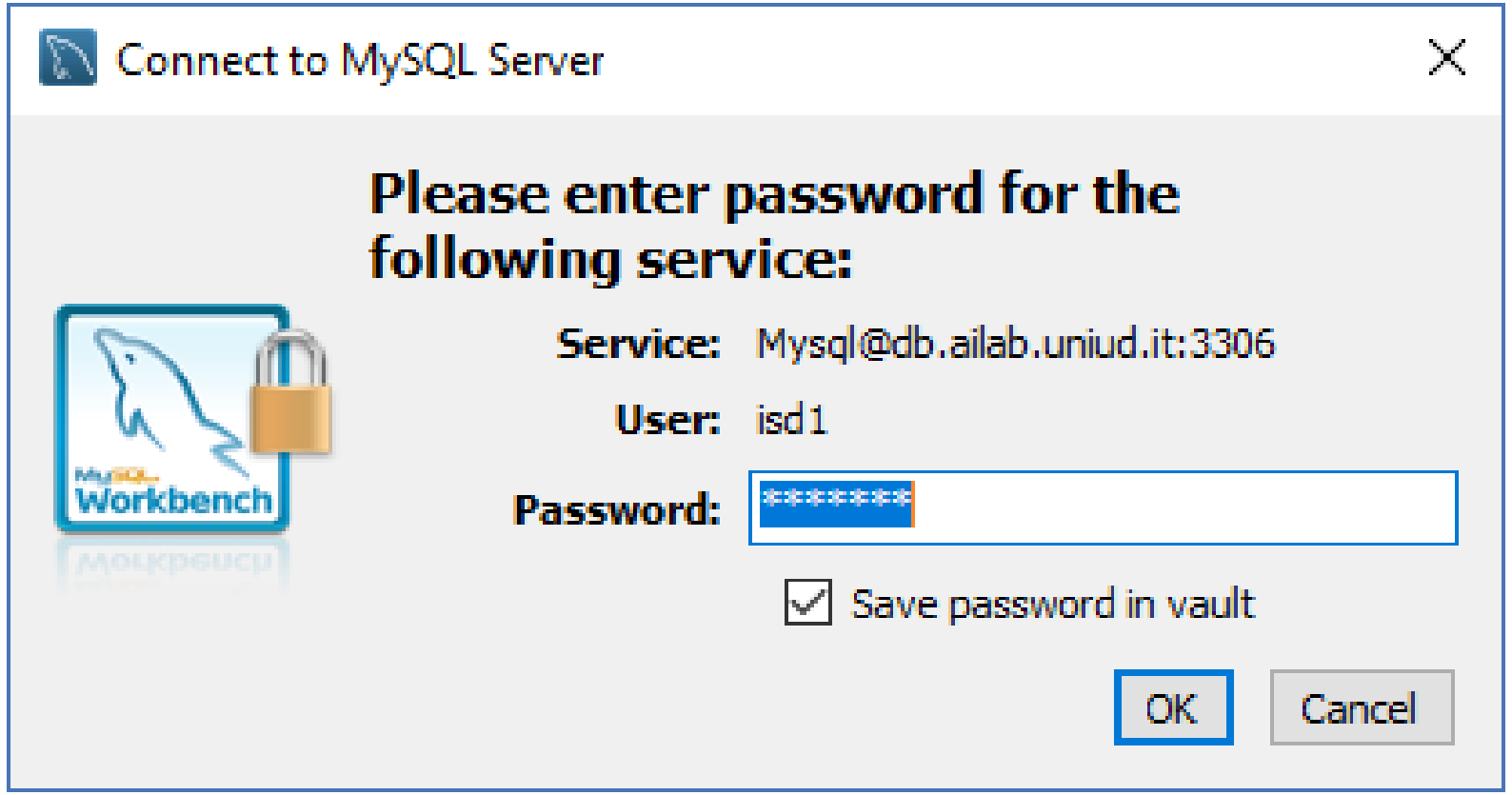
Step 1: click on the + ("plus") icon near to "MySQL Connections"



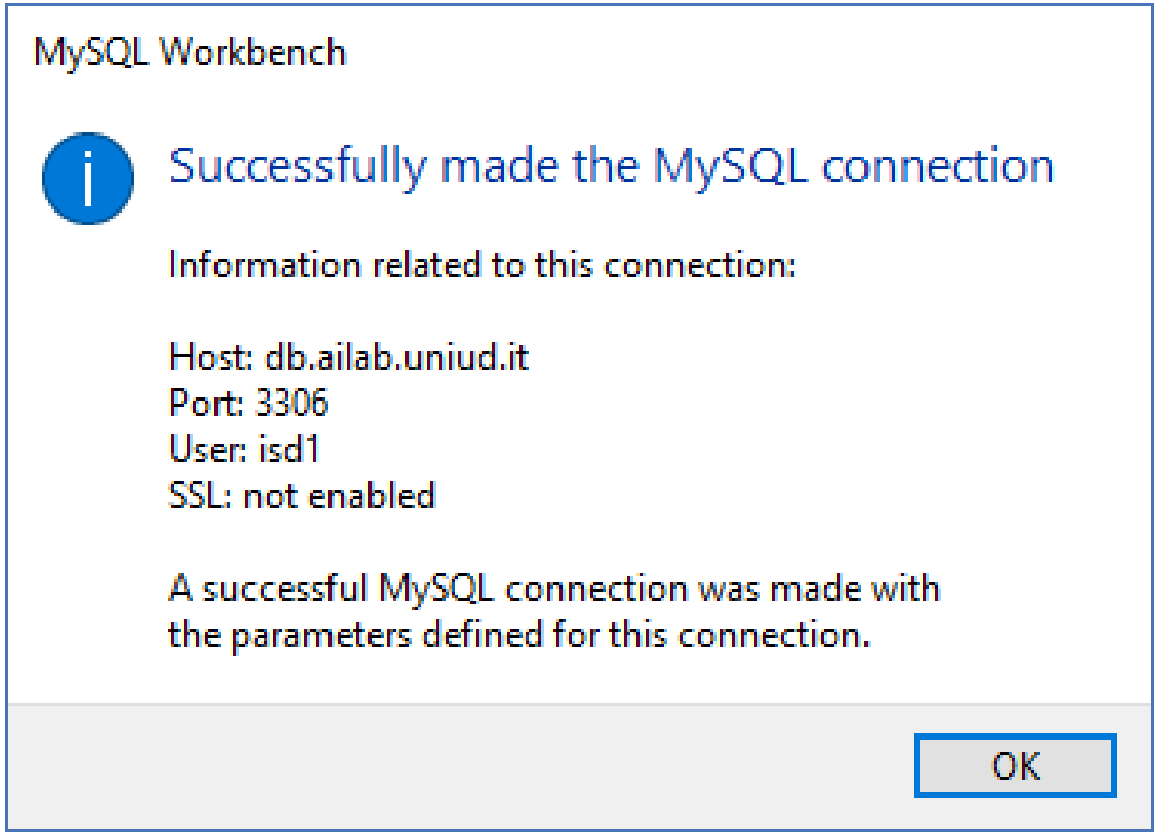
Step 2: insert the credentials (fields "Connection Name, Hostname, Username")



Step 3: click on "Test Connection"

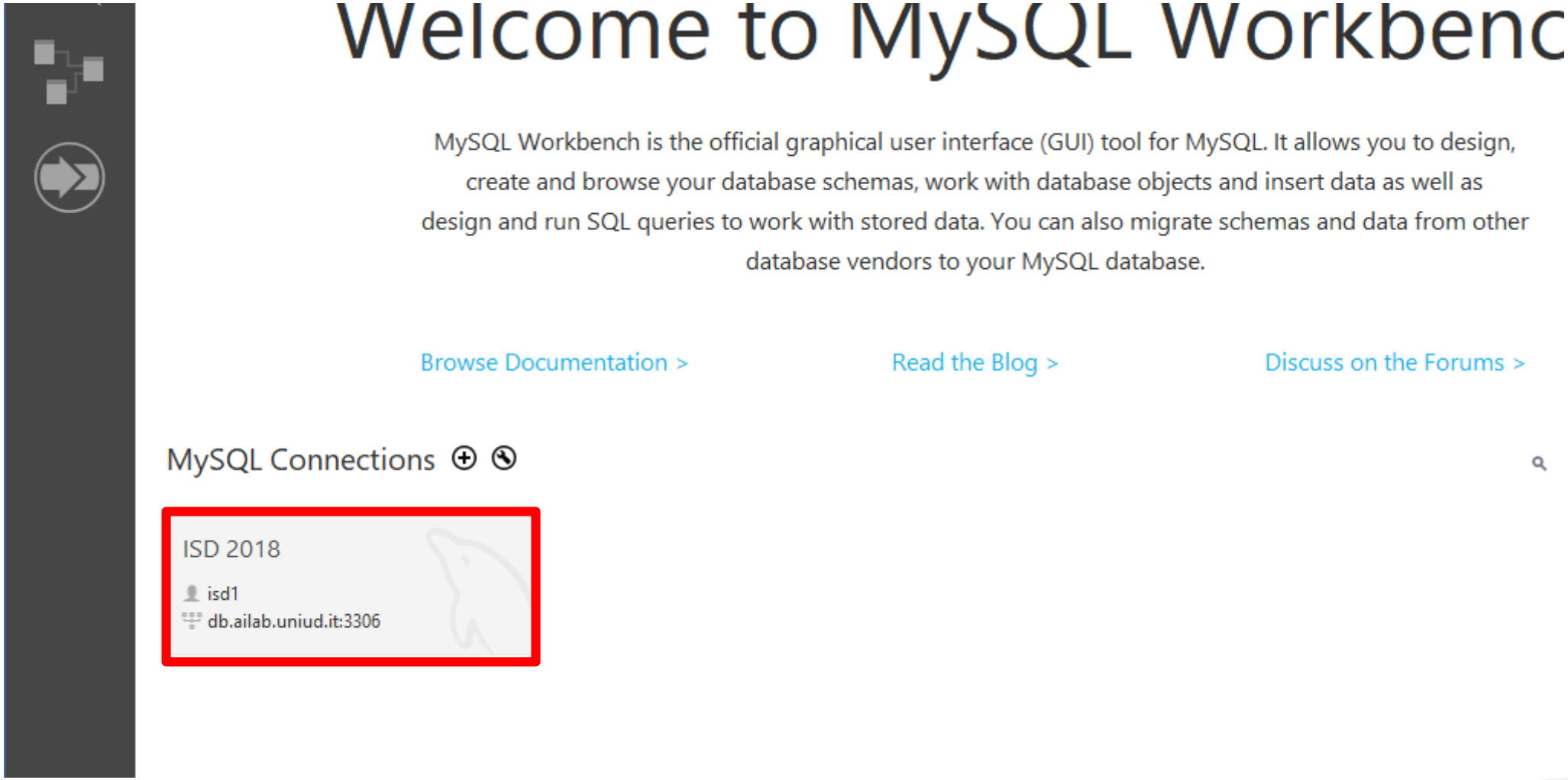


Step 4: Insert the password (look at the blackboard!)

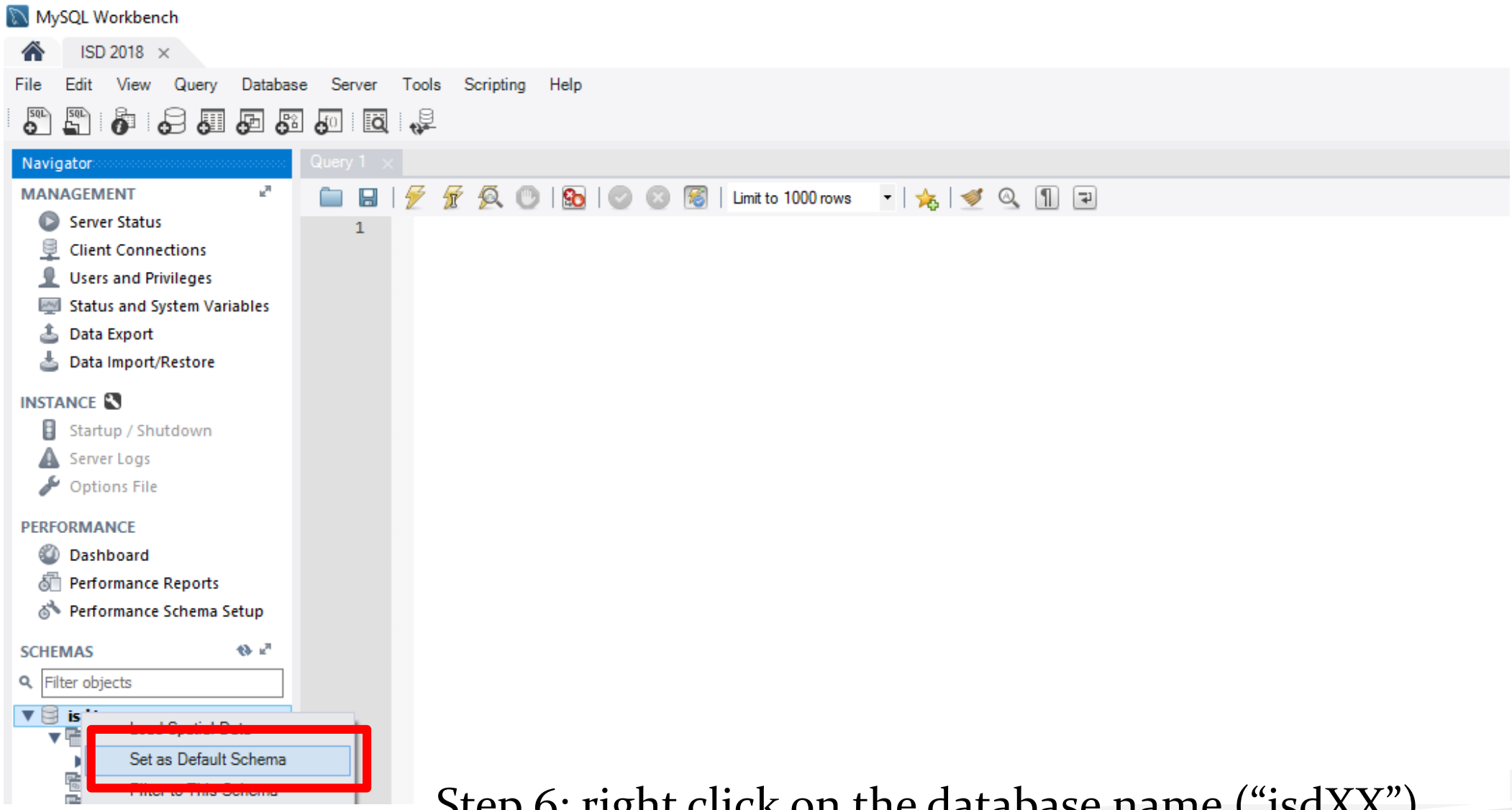


You should see **exactly** this window

If this window **doesn't show up** call me, I'll come and help you

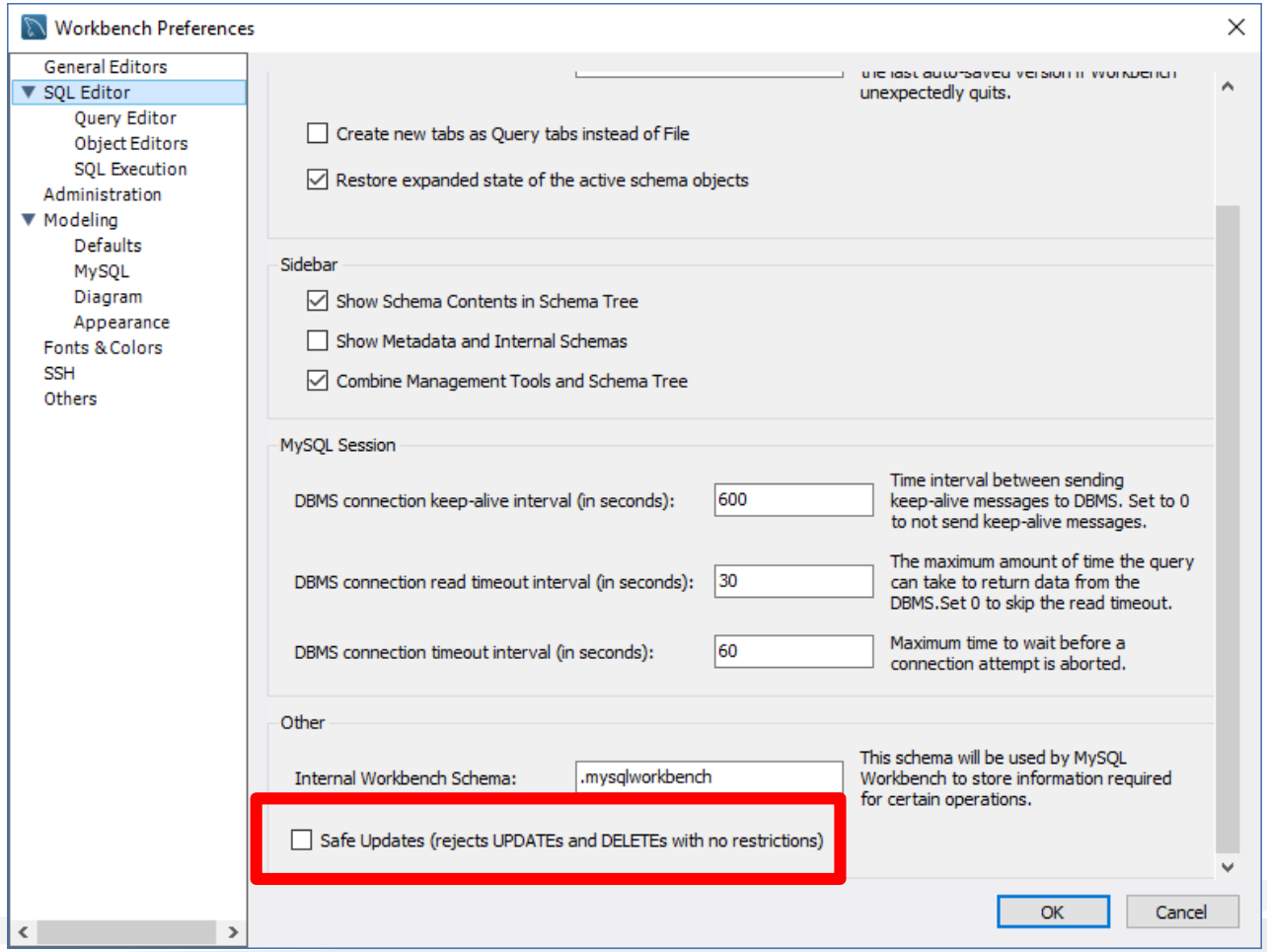


Step 5: double click on the newly created connection



Step 6: right click on the database name ("isdXX"), select "Set as Default Schema"

Disable safe mode



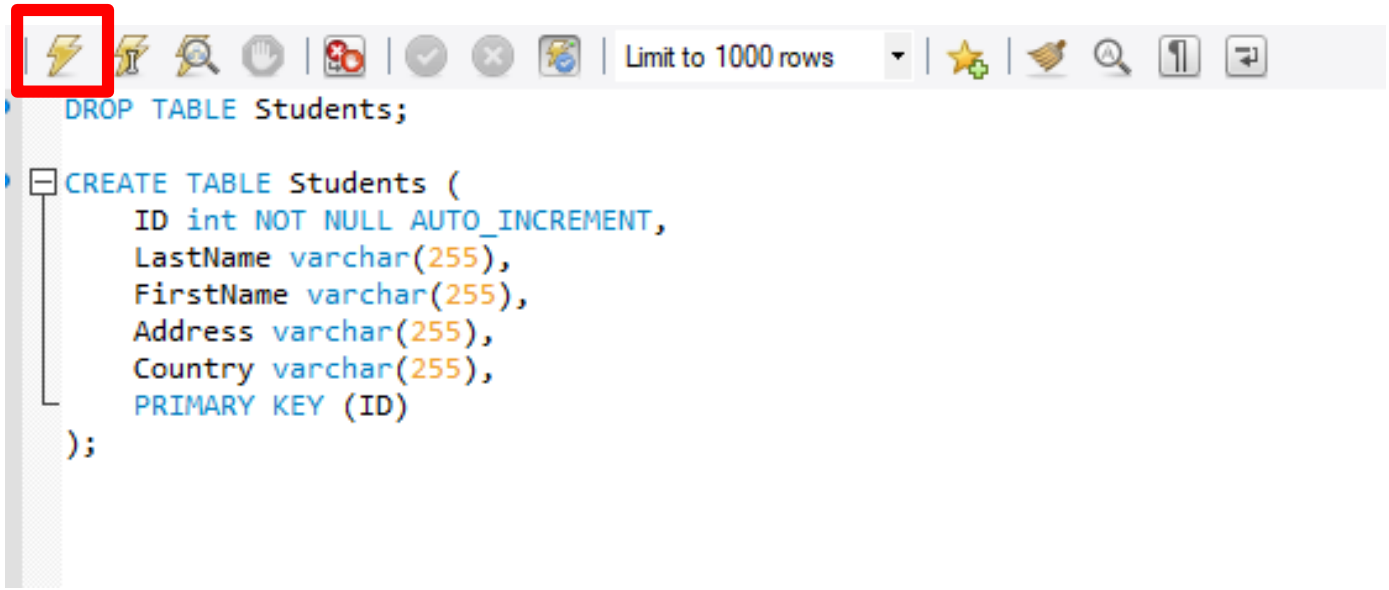
- Click on “Edit → Preferences”
- Click on “SQL Editor” in the left panel
- Scroll down and disable “Safe Updates”

Our first table

```
CREATE TABLE Students (
    ID int NOT NULL AUTO_INCREMENT,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    Country varchar(255),
    PRIMARY KEY (ID)
);
```

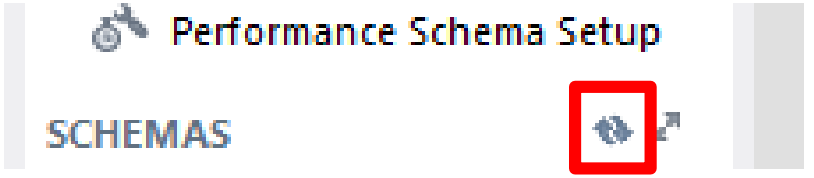
Our first table/2

- Copy/paste the code into the text area
- Click on the lightning icon to run it



Our first table/3

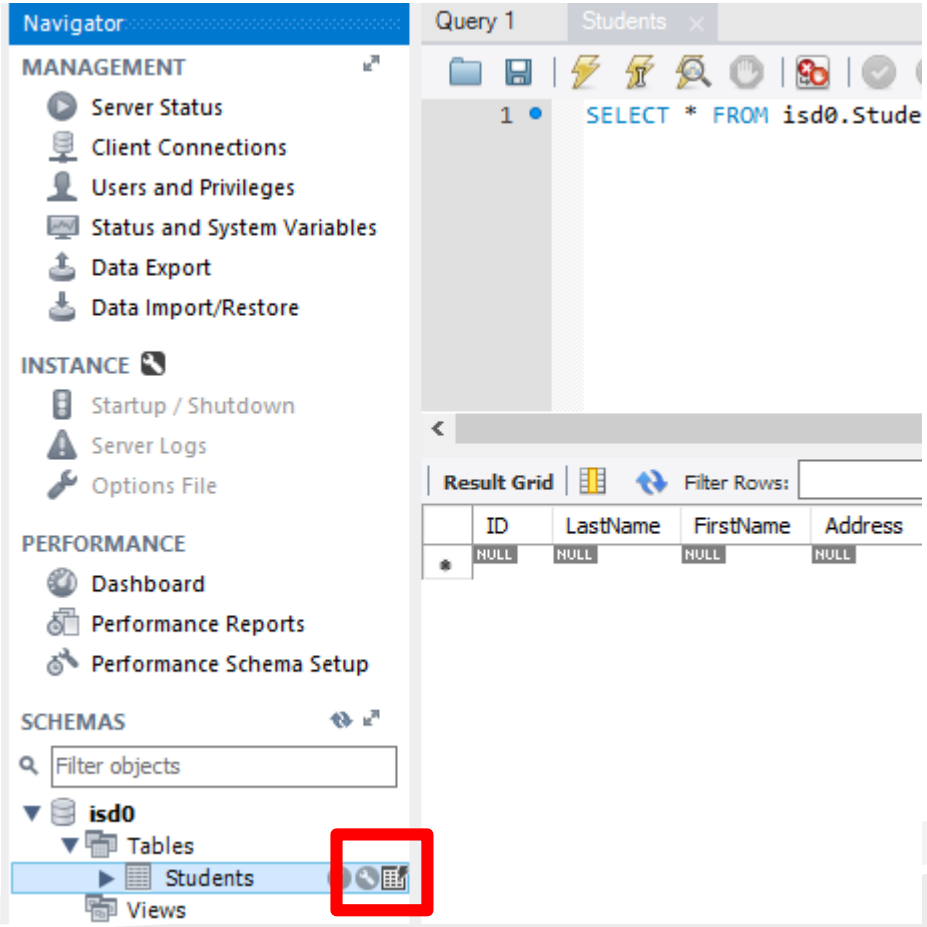
- Click on the refresh icon near “Schemas”: the table should appear



Our first table/4

To see the table:

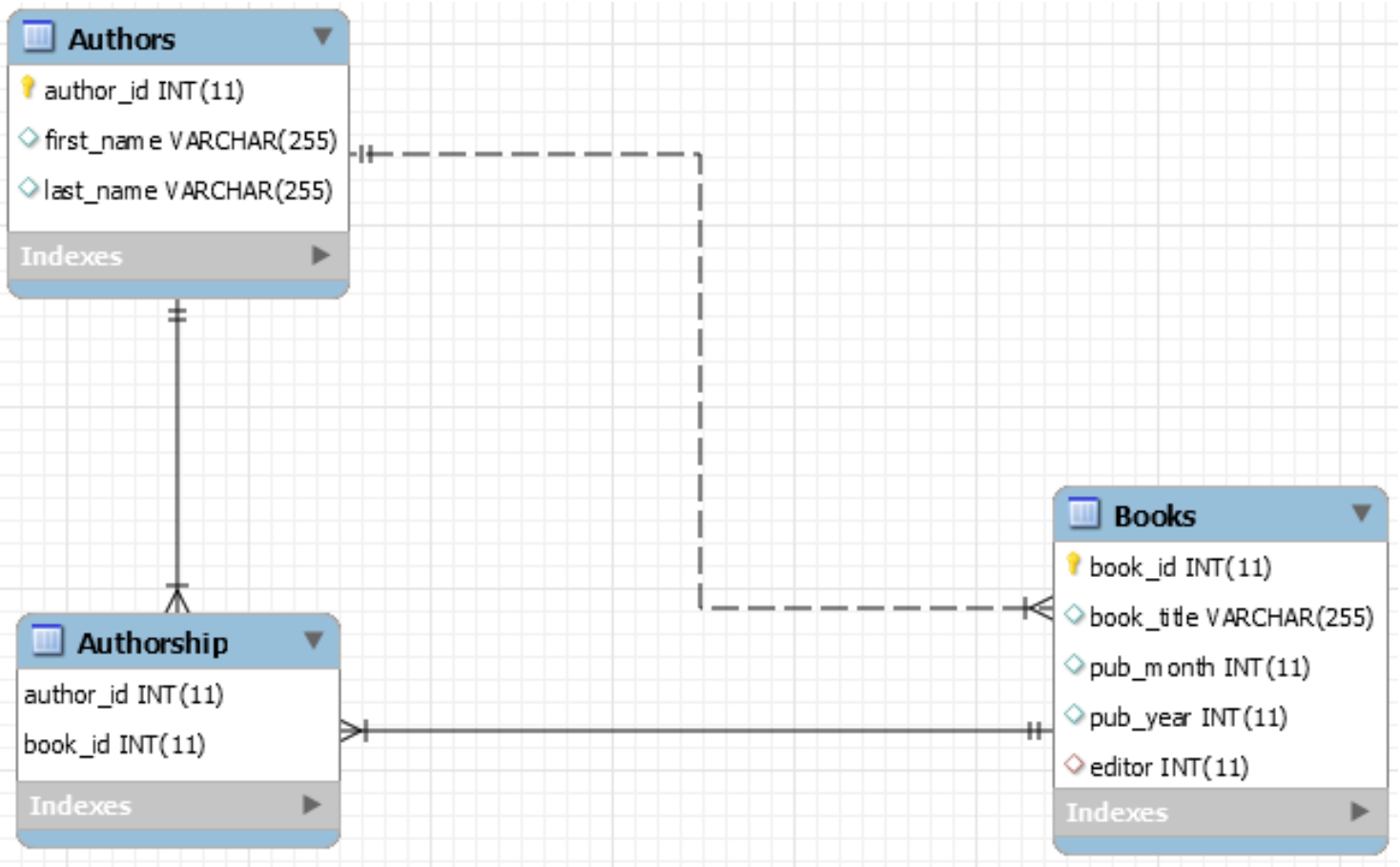
- Open “isdXX”
- Open “tables”
- Click on the table icon with the small lightning; this will automatically show all the contents of the table.



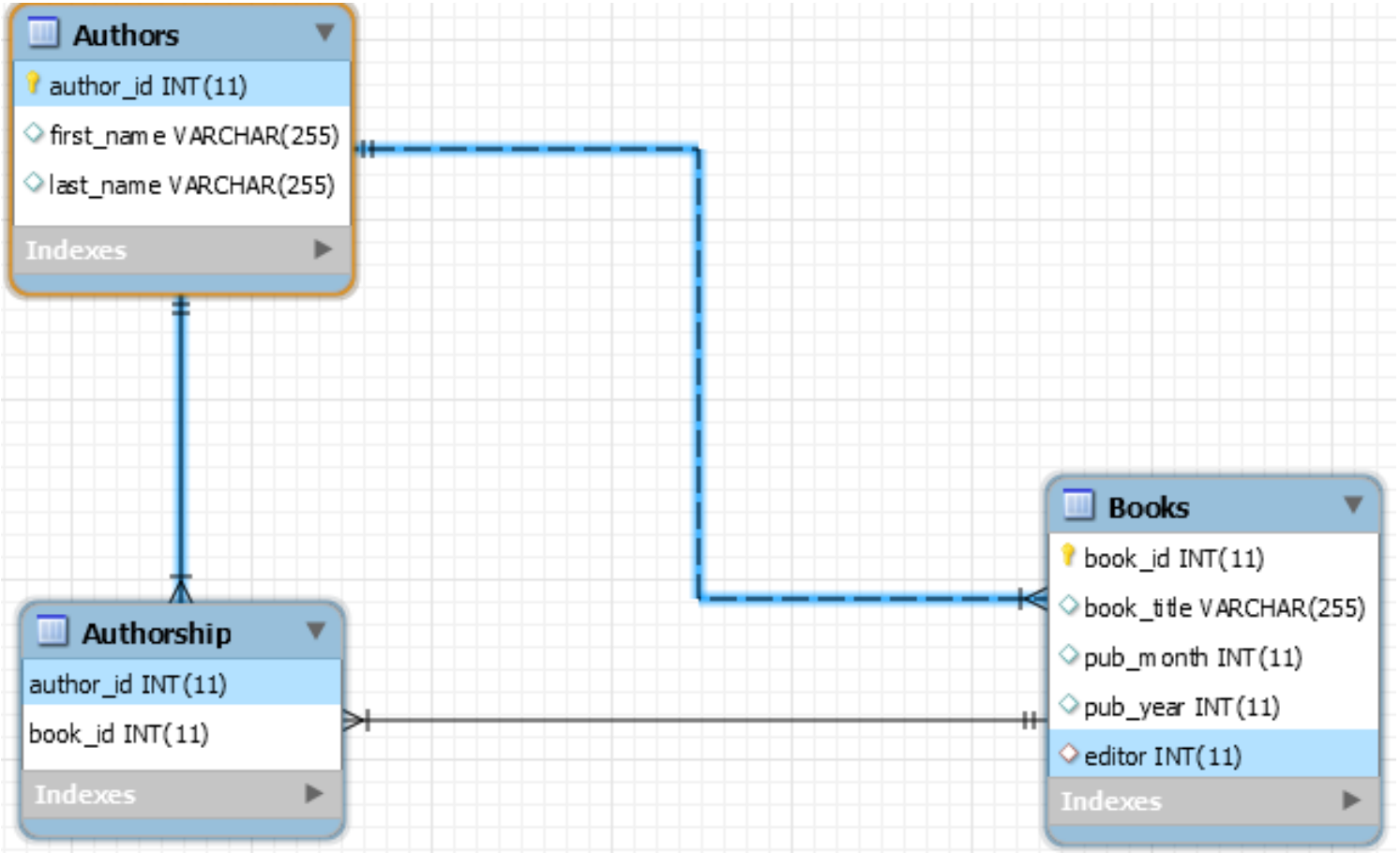
Hands-on exercises

Example: a book database

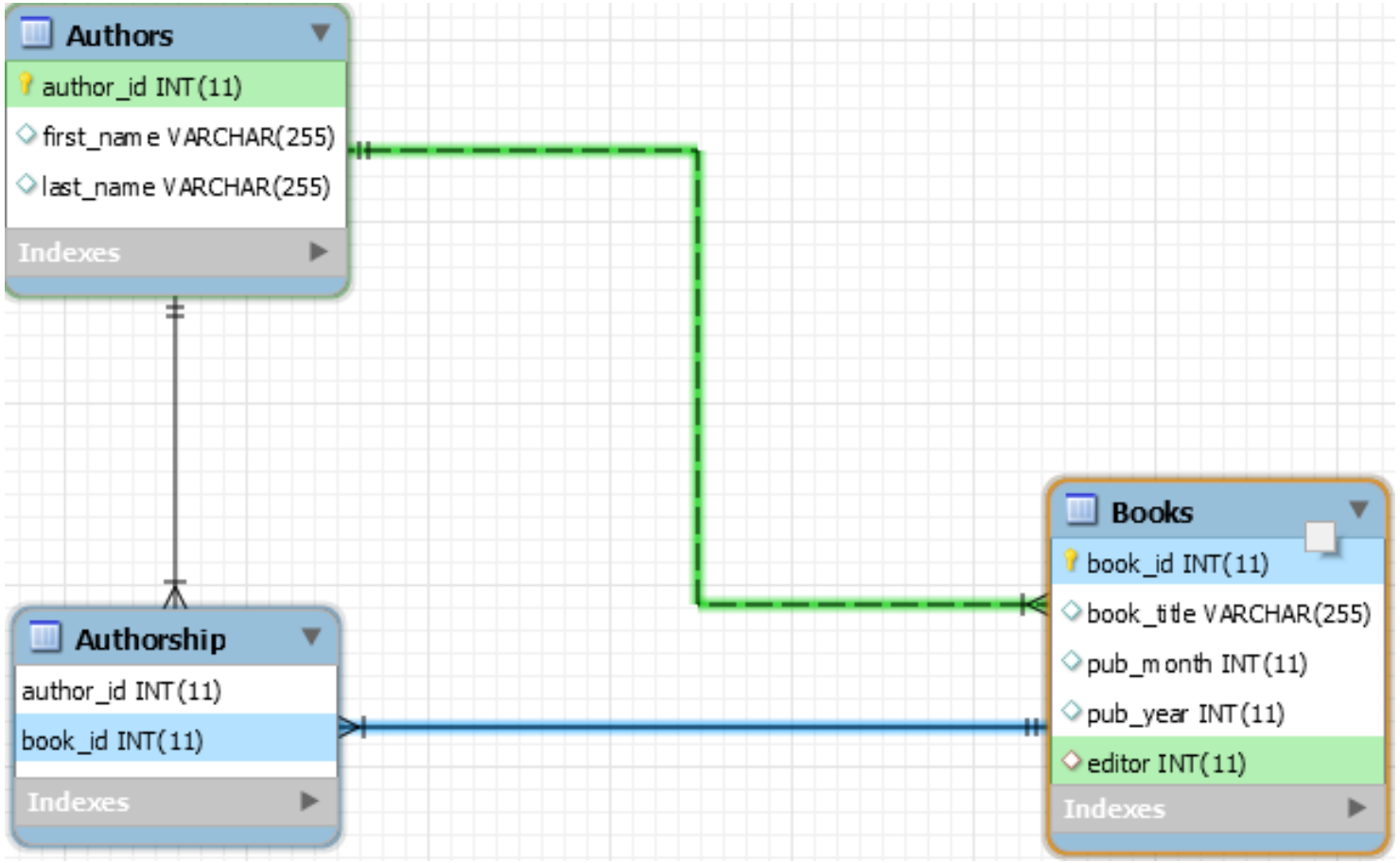
- We have to keep track of the records for a library.
- A book is identified by an ID, and its attributes are the title, the publication month and year, and its editor.
- A book can have more than one author.
- A book must have exactly one editor.
- An author/editor is identified by an ID, and its attributes are her/his name and surname.



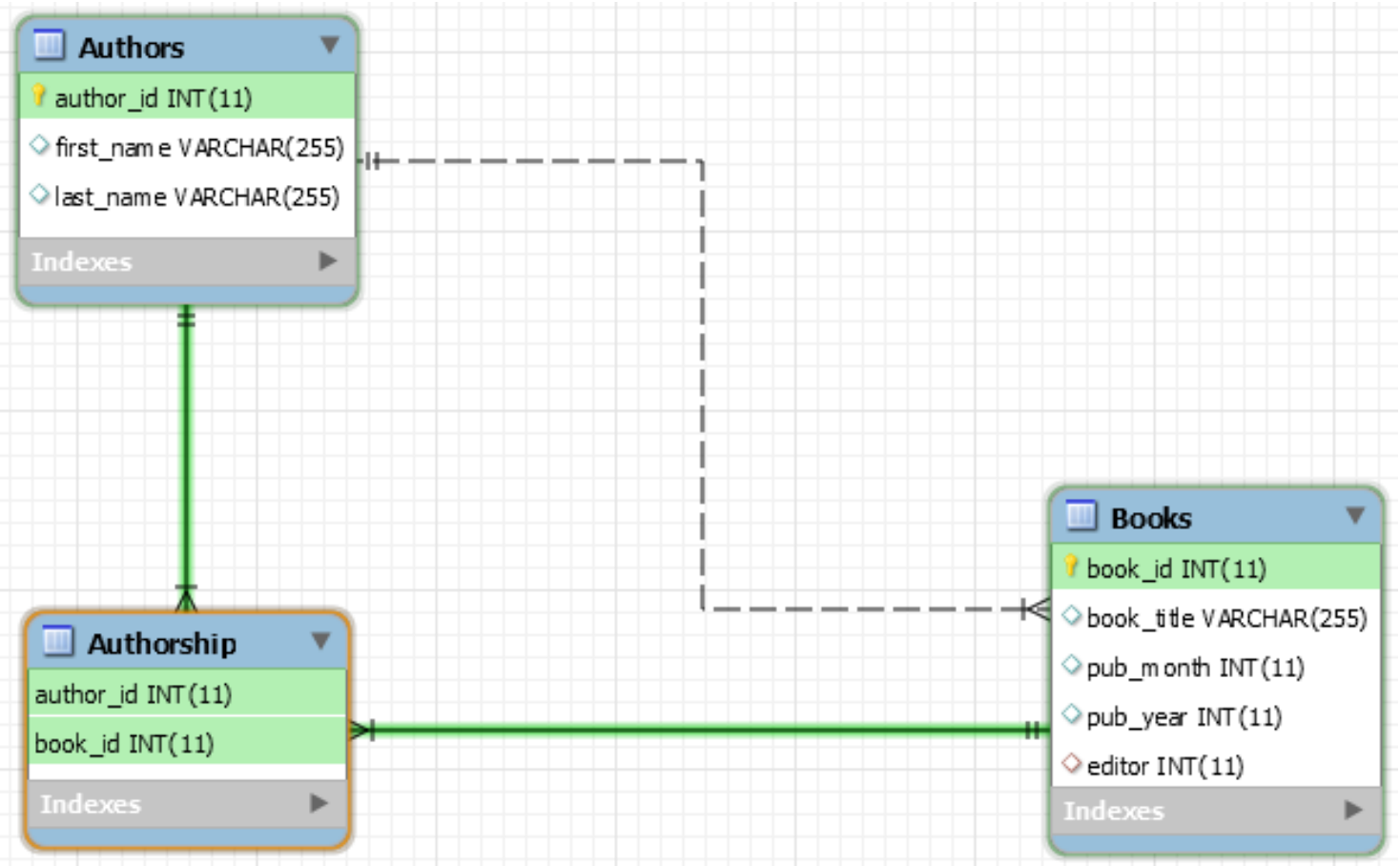
Tables: **Authors**, **Books**, **Authorship**



`author_id`: key for **`Authors`**
Referenced by `Authorship.editor` and by `Books.editor`



`book_id`: key for **`Books`**
Referenced by `Authorship.book_id`



`Authorship`: the key is **the couple** `author_id`, `book_id`

Create the toy database

- Draw the ER diagram (optional).
- Download the code for the first day from **`https://github.com/basaldella/isd2018`** and run it.
- Now you should have the three tables populated with some data: view it by running

```
SELECT * FROM Books
```

  Repeat the selection for the other two tables.

Exercises: 2

1. View the content of all the tables (`SELECT * …`)
2. Find all the books published in 1956.
3. Find the titles of the books edited by Dimitri Kent.
4. Sort the books by publication year.
5. Find the surnames of the authors who wrote at least a book.
6. Sort in descending order the authors by number of books authored and edited.

Exercises: hard

7. Find the authors who did not publish or author a book.
8. Find the authors who are publishers and editors.