

SE212 Database system and design

Agenda

- MYSQL and mysql workbench
- DDL in action and intro to DML
- Project Proposal is due by Sunday midnight

Recommend for project

- Start wit HTML5, JS and CSS
- If you think you already master them
- Try small framework, e.g. [play](#), [grails](#) which are based on MVC patterns
- If you want to try PHP, recommend cakePHP
- Next, you will learn revision control (Advanced programming),
Interactive Design (Interactive web development)

Workbench

Download and install

Download and install Workbench

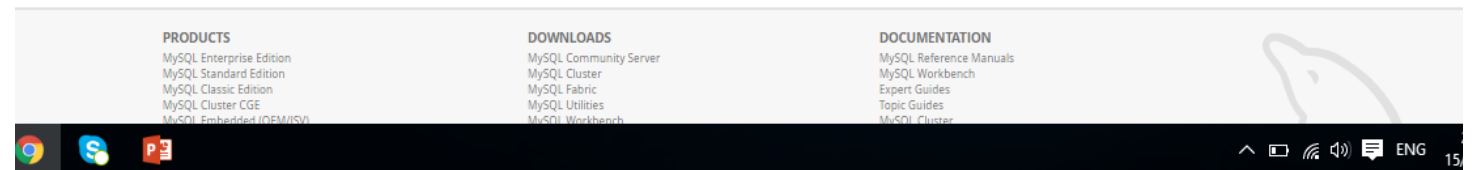
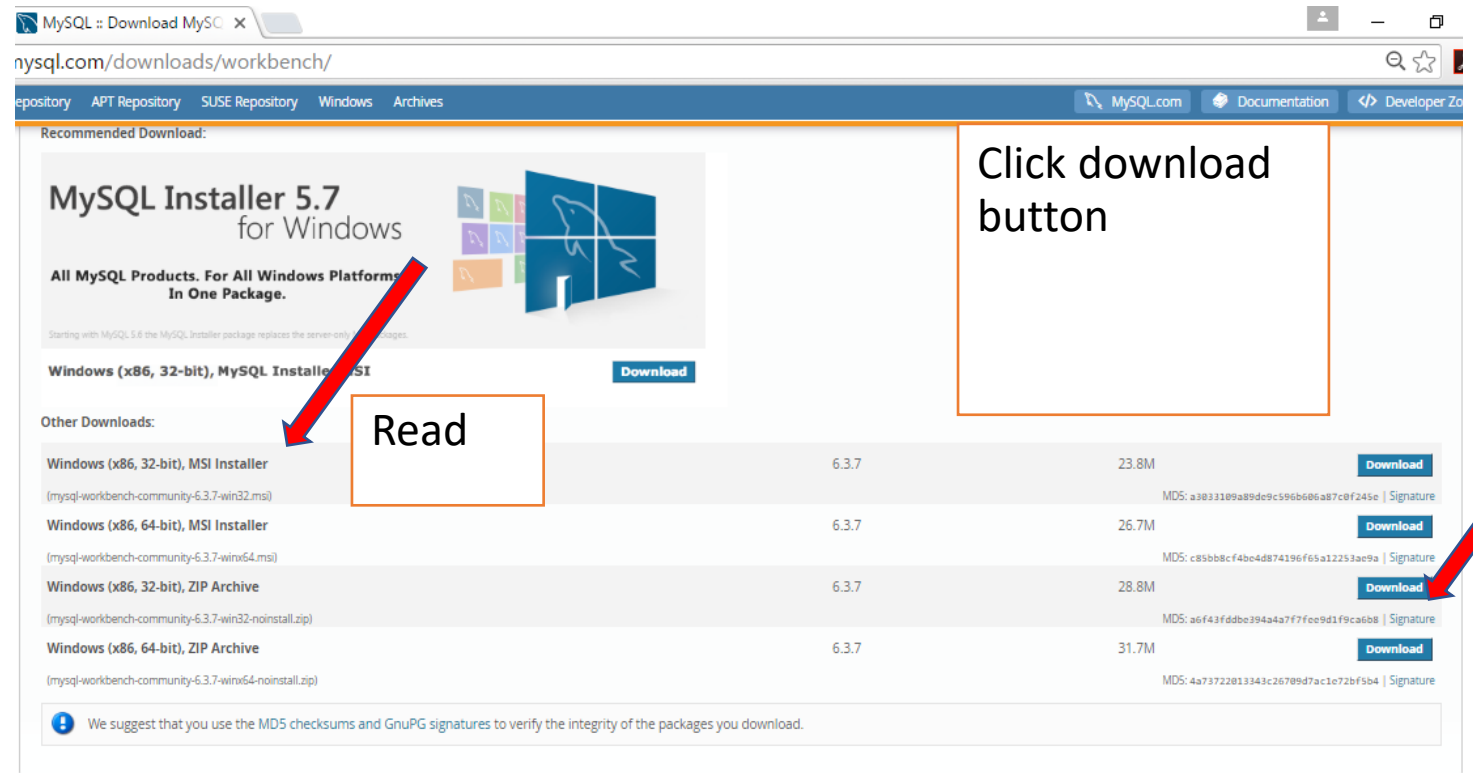
google.com

- Type Download Workbench

- Select

<https://dev.mysql.com/downloads/workbench/>


- Start download
- 32 or 64 bit



SQL CREATE TABLE Statem

MySQL :: Begin Your Downr

← → ↺ 🔒 https://dev.mysql.com/downloads/file/?id=463709 🔍 ☆ 📄 ☰



The world's most popular open source database

MySQL.com Downloads Documentation Developer Zone

Enterprise Community Yum Repository APT Repository SUSE Repository Windows Archives

MySQL on WindowsMySQL Yum RepositoryMySQL APT RepositoryMySQL SUSE RepositoryMySQL Community ServerMySQL ClusterMySQL FabricMySQL RouterMySQL UtilitiesMySQL ShellMySQL WorkbenchMySQL ConnectorsOther Downloads

Begin Your Download - mysql-workbench-community-6.3.7-winx64-noinstall.zip

Login Now or Sign Up for a free account.
An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

Click

Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

mysql-workbench-community-6.3.7-winx64-noinstall.zip requires Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

CONTACT SALES

USA: +1-866-221-0634
Canada: +1-866-221-0634

Germany: +49 89 143 01280
France: +33 1 57 60 83 57
Italy: +39 02 249 59 120

PRODUCTS


MySQL Enterprise Edition
MySQL Standard Edition
MySQL Classic Edition
MySQL Cluster CGE
MySQL Embedded (OEM/ISV)


DOWNLOADS


MySQL Community Server
MySQL Cluster
MySQL Fabric
MySQL Utilities
MySQL Workbench

DOCUMENTATION

MySQL Reference Manuals
MySQL Workbench
Expert Guides
Topic Guides
MySQL Cluster

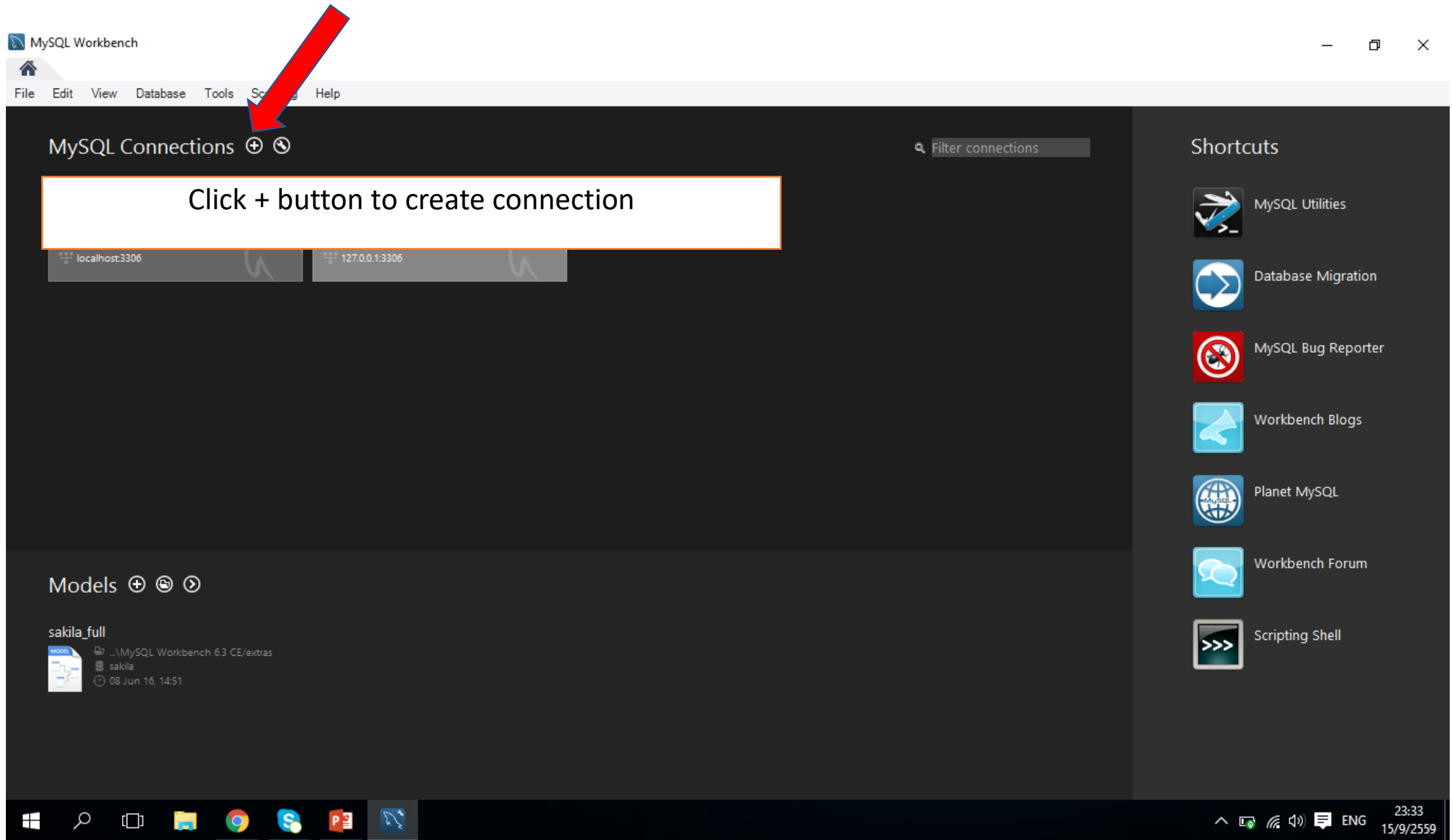






23:25
15/9/2559

Create Connection



MySQL Connections + -

Local instance mysql

root
localhost:3306

Models + - >

sakila_full

..\\MySQL Workbench 6.3 CE\\extras
sakila
08 Jun 16, 14:51

1.Type Connection Name

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave

2.Click

Configure Server Management...

Test Connection

Cancel

OK

Shortcuts



MySQL Utilities



Database Migration



MySQL Bug Reporter



Workbench Blogs



Planet MySQL



Workbench Forum



Scripting Shell





MySQL Connections + -

Local instance mysql

root
localhost:3306

Models + - >

sakila_full

...\\MySQL Workbench 6.3 CE\\extras
sakila
08 Jun 16, 14:51

Setup New Connection

Connection Name: firstCon

Type a name for the connection

Connection Method: Standard (TCP/IP)

Method to use to connect to the RDBMS

Parameters

Hostname: 127.0.0.1

Username: root

Password:

Default Schema:

MySQL Workbench



Successfully made the MySQL connection

Information related to this connection:

Host: 127.0.0.1
Port: 3306
User: root
SSL: not enabledA successful MySQL connection was made with
the parameters defined for this connection.

OK

1.Click

2.Click

Configure Server Management...

Test Connection

Cancel

OK

Shortcuts



MySQL Utilities



Database Migration



MySQL Bug Reporter



Workbench Blogs



MySQL



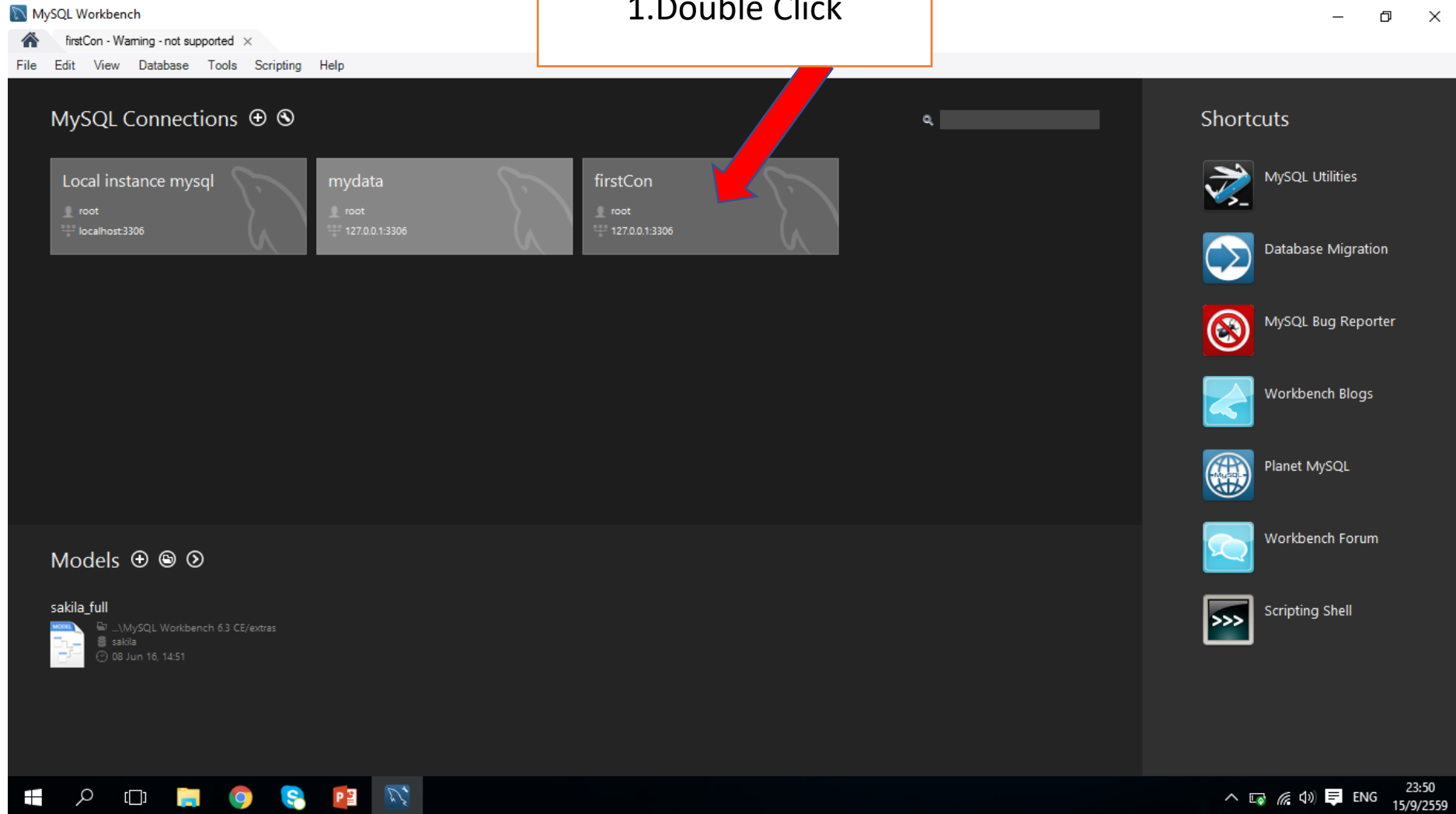
Workbench Forum



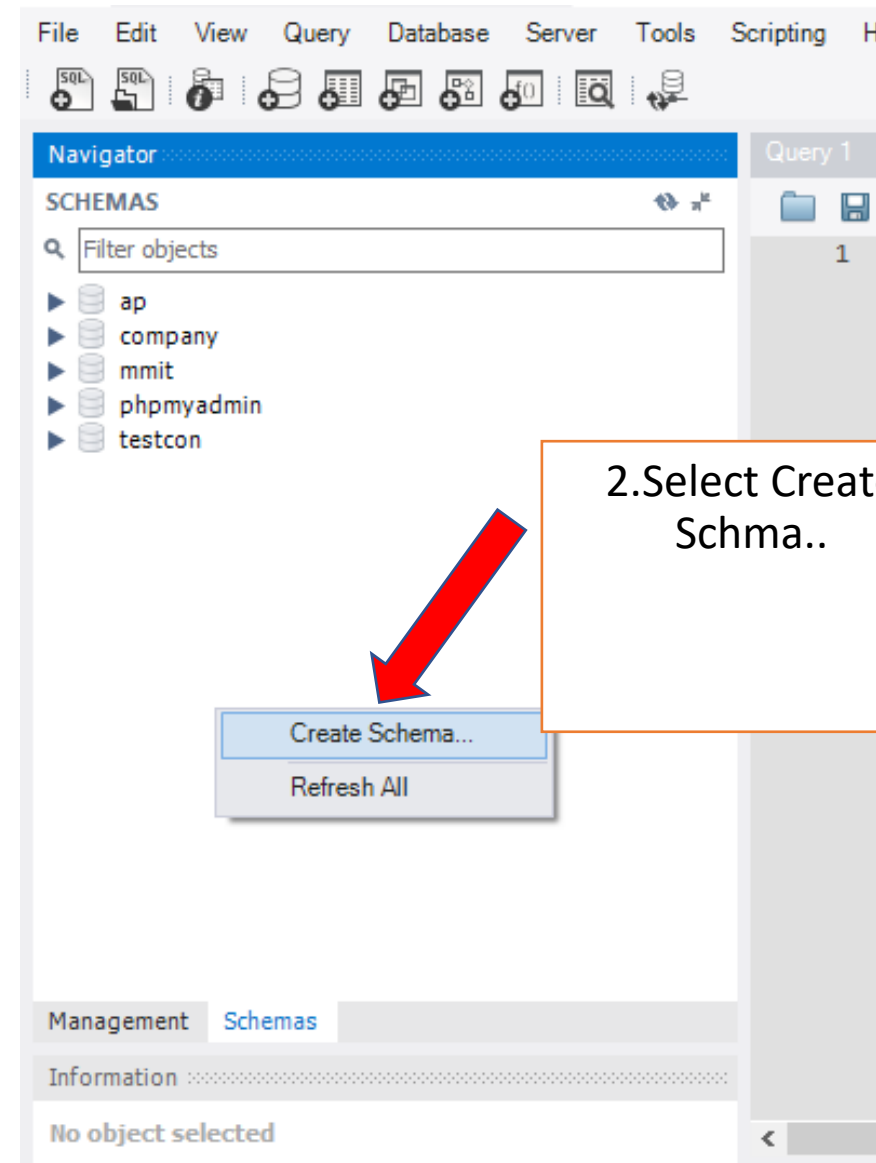
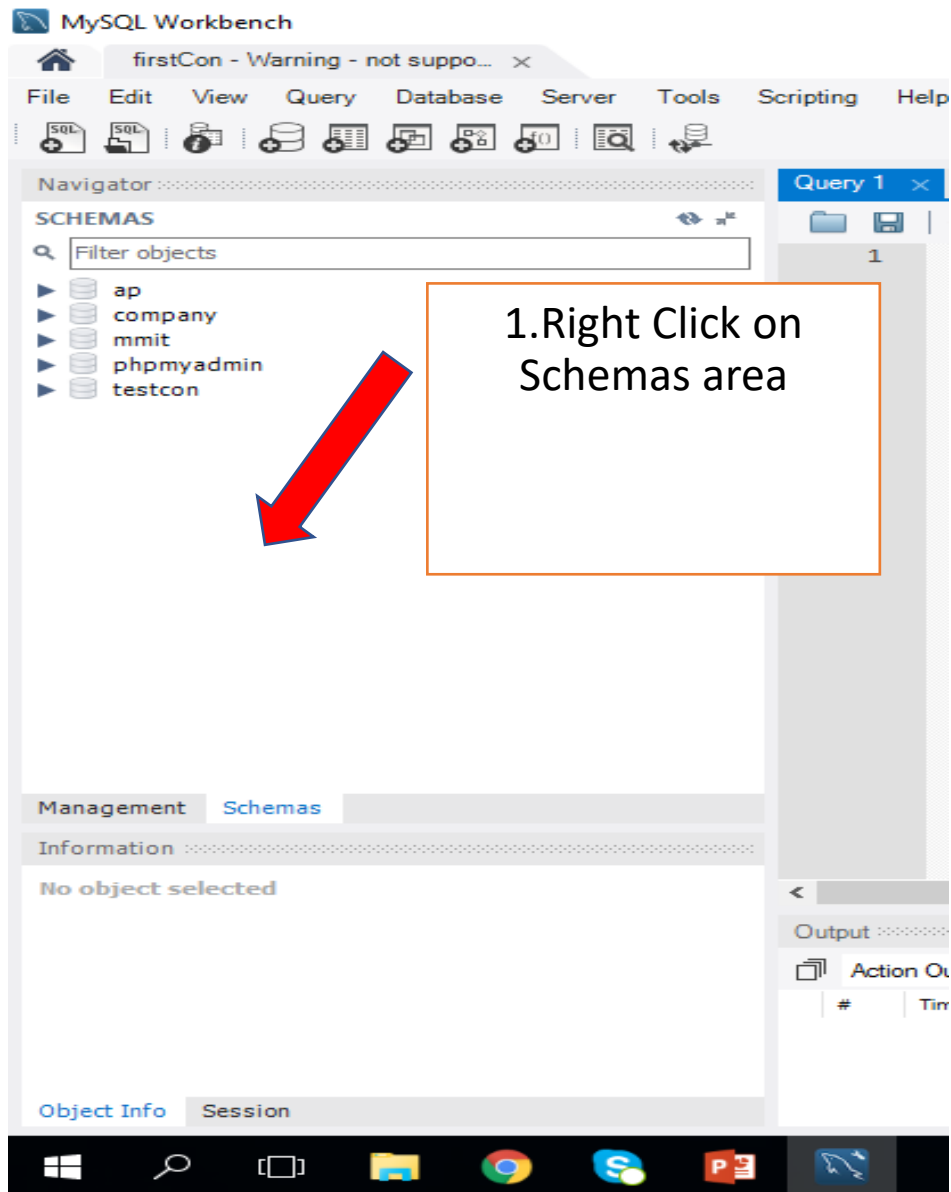
Scripting Shell



1.Double Click



Create Schema



MySQL Workbench

firstCon - Warning - not suppo... x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- ap
- company
- mmit
- phpmyadmin
- testcon

Management Schemas

Information

No object selected

Object Info Session

Query 1 first_schema - Schema x

Name: first_schema

Rename References

Collation: utf8 - utf8_general_ci

The name of the schema. It is recommended to use only alpha-numeric characters. Spaces should be avoided and be Refactor model, changing all references found in view, triggers, stored procedures and functions from the old

Specifies which charset/collations the schema's tables will use if they do not have an explicit setting. Common choices

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Schema

Apply Revert

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

1.Type Schma name

2.select utf8-utf8_general_ci

0:00 16/9/2559

MySQL Workbench

firstCon - Warning - not suppo... x

File Edit View Query Database Server Tools Scripting Help

Apply SQL Script to Database

Navigator

SCHEMAS

Filter objects

- ap
- company
- mmit
- phpmymadmin
- testcon

Management Schemas

Information

No object selected

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

```
1 CREATE SCHEMA `first_schema` DEFAULT CHARACTER SET utf8 ;
2
```

Back Apply Cancel

1 00:02:49 Apply changes to first_schema

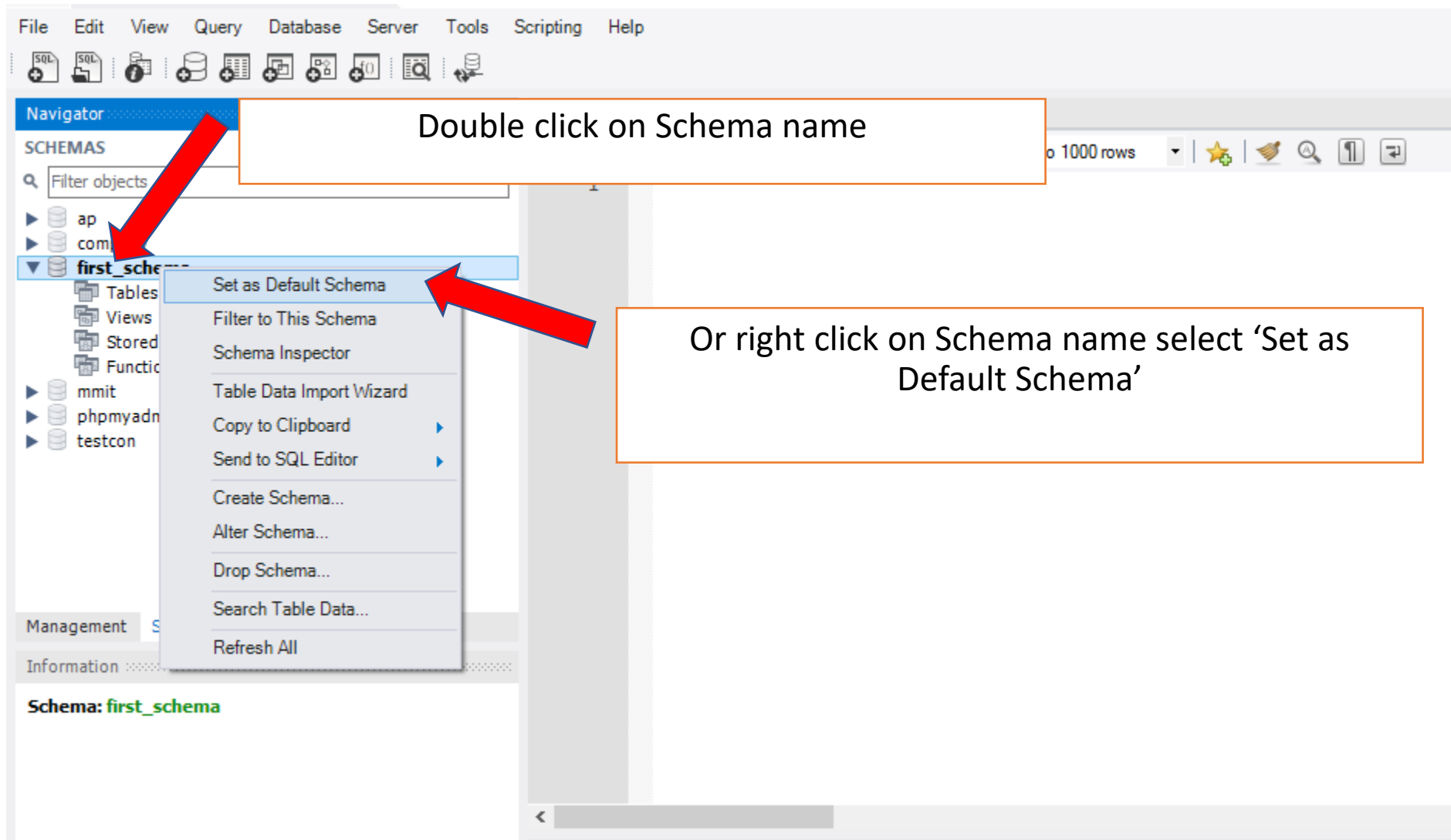
Preparing...

Duration / Fetch

Object Info Session

1. Read the SQL Script

2. Click Apply button



File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- ap
- company
- first_schema**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- mmit
- phpmyadmin
- testcon

Query 1

Limit to 1000 rows

```
1 CREATE TABLE Persons
2 (
3   PersonID int,
4   LastName varchar(255),
5   FirstName varchar(255),
6   Address varchar(255),
7   City varchar(255)
8 );
9
```

3. Read the Action Output

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	00:02:49	Apply changes to first_schema	Changes applied	
2	00:04:30	Apply changes to first_schema	No changes detected	
3	00:09:12	CREATE TABLE Persons (PersonID int, LastName varchar(255), FirstName varchar...	0 row(s) affected	0.016 sec

CREATE TABLE

Scripting Help

Query 1 q3

```
1 CREATE TABLE Persons
2 (
3   PersonID int,
4   LastName varchar(255),
5   FirstName varchar(255),
6   Address varchar(255),
7   City varchar(255)
8 );
9
```

• Columns

File Edit View Query Database Server Tools Scripting

Navigator

SCHEMAS

Filter objects

- ap
- company
- first_schema**
 - Tables
 - persons**
 - Views
 - Stored Procedures
 - Functions
- mmit
- phpmyadmin
- testcon

Management Schemas

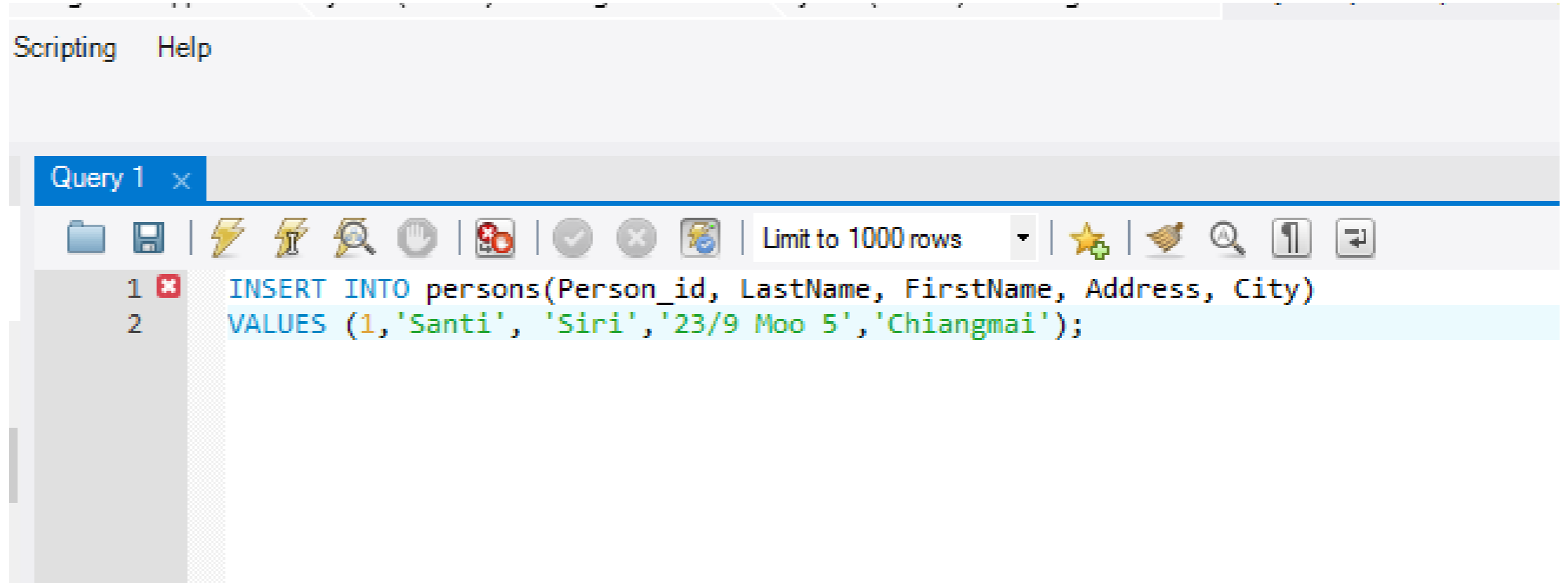
Information

Table: persons

Columns:

PersonID	int(11)
LastName	varchar(255)
FirstName	varchar(255)
Address	varchar(255)
City	varchar(255)

INSERT INTO Statement



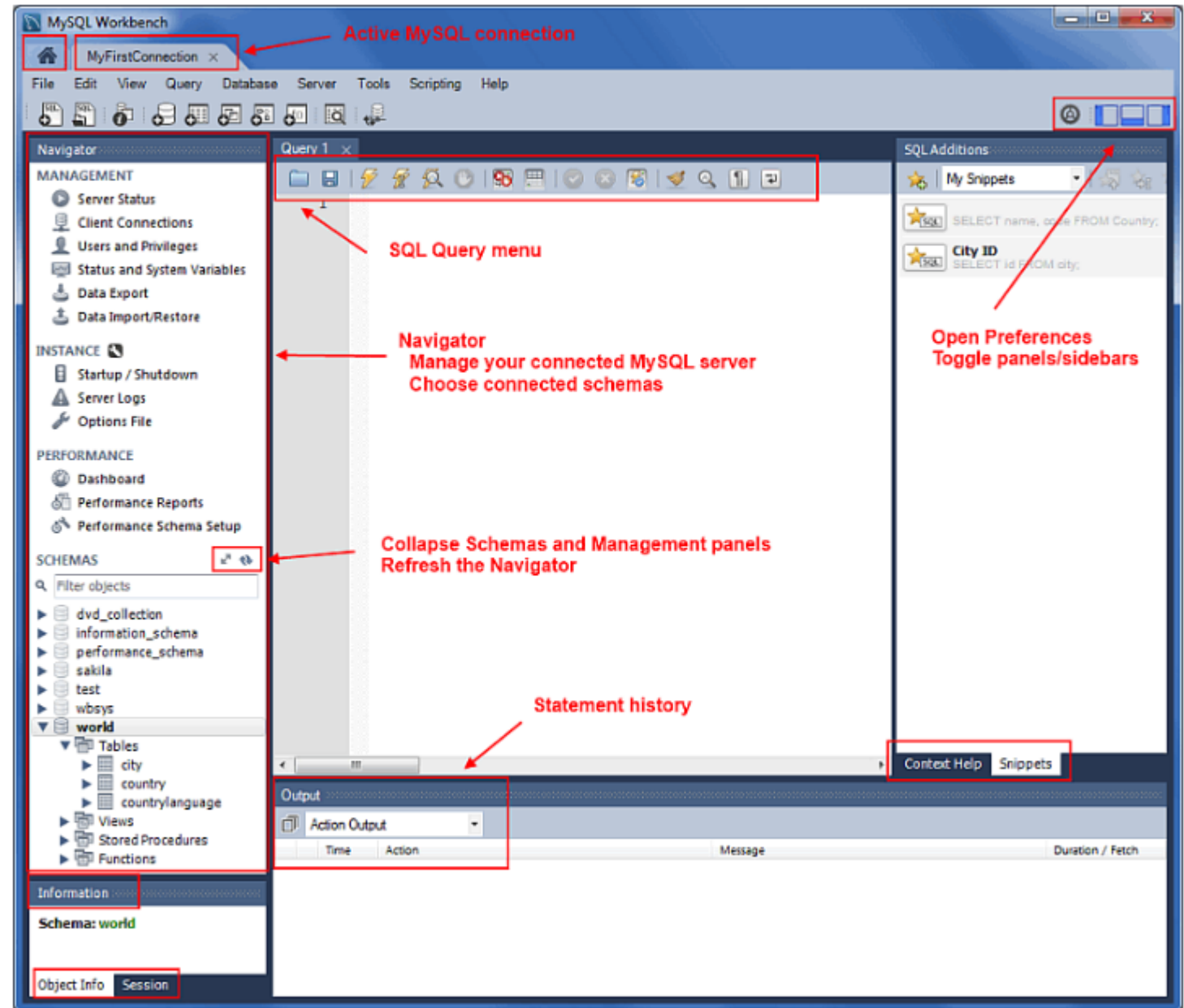
SELECT Statement

The screenshot shows a SQL IDE interface. At the top, there are tabs for 'Query 1' and 'SQL File 3*'. Below the tabs is a toolbar with various icons for file operations, execution, and settings. A dropdown menu shows 'Limit to 1000 rows'. The main text area contains the SQL statement: `SELECT * FROM persons`. Below the text area is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid itself is a table with the following data:

	PersonID	LastName	FirstName	Address	City
▶	1	Santi	Siri	23/9 Moo 5	Chiangmai

On the right side of the interface, there is a sidebar with a 'Result Grid' button (highlighted in blue) and a 'Form' button. Above these buttons, there is a partially visible text: 'SQLAdditio' and 'Automa to manu or to to'.

Figure 8.1 SQL Editor GUI



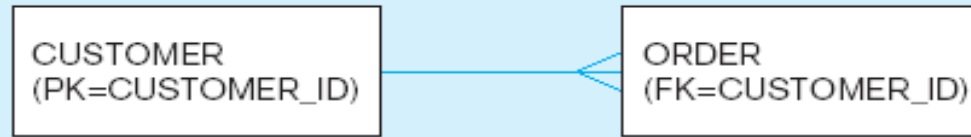
Sample database

- Use Employee database
- Reference:
- <https://dev.mysql.com/doc/employee/en/>
- https://github.com/datacharmer/test_db

Data Integrity Controls

- **Referential integrity**—constraint that ensures that foreign key values of a table must match primary key values of a related table in 1:M relationships
- Restricting:
 - **Deletes** of primary records
 - **Updates** of primary records
 - **Inserts** of dependent records

Figure 7-7 Ensuring data integrity through updates



Restricted Update: A customer ID can only be deleted if it is not found in ORDER table.

```
CREATE TABLE CUSTOMER_T
(CUSTOMER_ID      INTEGER DEFAULT 'C999' NOT NULL,
 CUSTOMER_NAME    VARCHAR(40)          NOT NULL,
```

...

```
CONSTRAINT CUSTOMER_PK PRIMARY KEY (CUSTOMER_ID),
ON UPDATE RESTRICT);
```

Cascaded Update: Changing a customer ID in the CUSTOMER table will result in that value changing in the ORDER table to match.

```
... ON UPDATE CASCADE);
```

Set Null Update: When a customer ID is changed, any customer ID in the ORDER table that matches the old customer ID is set to NULL.

```
... ON UPDATE SET NULL);
```

Set Default Update: When a customer ID is changed, any customer ID in the ORDER tables that matches the old customer ID is set to a predefined default value.

```
... ON UPDATE SET DEFAULT);
```

Relational
integrity is
enforced via
the primary-
key to foreign-
key match

Customer_T						
CustomerID	CustomerName	CustomerAddress	CustomerCity	CustomerState	CustomerPostalCode	
1	Contemporary Casuals	1355 S Hines Blvd	Gainesville	FL	32601-2871	
2	Value Furniture	15145 S.W. 17th St.	Plano	TX	75094-7743	
3	Home Furnishings	1900 Allard Ave.	Albany	NY	12209-1125	
4	Eastern Furniture	1925 Beltline Rd.	Carteret	NJ	07008-3188	
5	Impressions	5585 Westcott Ct.	Sacramento	CA	94206-4056	
6	Furniture Gallery	325 Flatiron Dr.	Boulder	CO	80514-4432	
7	Period Furniture	394 Rainbow Dr.	Seattle	WA	97954-5589	
8	California Classics	816 Peach Rd.	Santa Clara	CA	96915-7754	
9	M and H Casual Furniture	3701 E. 1st St.	San Jose	CA	95120-2314	
10	Seminole Interiors	2401 N. 1st St.	Phoenix	AZ	85016-4473	
11	American Euro Lifestyles	2421 N. 1st St.	Phoenix	AZ	85016-5621	
12	Battle Creek Furniture	345 N. 1st St.	Phoenix	AZ	85015-3401	
13	Heritage Furnishings	667 N. 1st St.	Phoenix	AZ	85013-8834	
14	Kaneohe Homes	112 N. 1st St.	Phoenix	AZ	85014-2537	
15	Mountain Scenes	413 N. 1st St.	Phoenix	AZ	85013-4432	
Record: 14 of 15						

Order_T		
OrderID	OrderDate	CustomerID
1001	10/21/2010	1
1002	10/21/2010	8
1003	10/22/2010	15
1004	10/22/2010	5
1005	10/24/2010	3
1006	10/24/2010	2
1007	10/27/2010	11
1008	10/30/2010	12
1009	11/5/2010	4
1010	11/5/2010	1
Record: 14 of 10		

OrderLine_T				
OrderID	ProductID	OrderedQuantity		
1001	1	2		
1001	2	2		
1001	4	1		
1002	3	5		
1003	3	3		
1004	6	2		
1004	8	2		
1005	4	4		
1006	4	1		
1006	5	2		
1006	7	2		
1007	1	3		
1007	2	2		
1008	3	3		
1008	8	3		
1009	4	2		
1009	7	3		
1010	8	10		
Record: 14 of 18				

Product_T				
ProductID	ProductDescription	ProductFinish	ProductStandardPrice	ProductLineID
1	End Table	Cherry	\$175.00	1
2	Coffe Table	Natural Ash	\$200.00	2
3	Computer Desk	Natural Ash	\$375.00	2
4	Entertainment Center	Natural Maple	\$650.00	3
5	Writers Desk	Cherry	\$325.00	1
6	8-Drawer Desk	White Ash	\$750.00	2
7	Dining Table	Natural Ash	\$800.00	2
8	Computer Desk	Walnut	\$250.00	3
Record: 14 of 8				

Delete Statement

- Removes rows from a table
- Delete certain rows
 - **DELETE FROM CUSTOMER_T WHERE STATE = 'HI';**
- Delete all rows
 - **DELETE FROM CUSTOMER_T;**

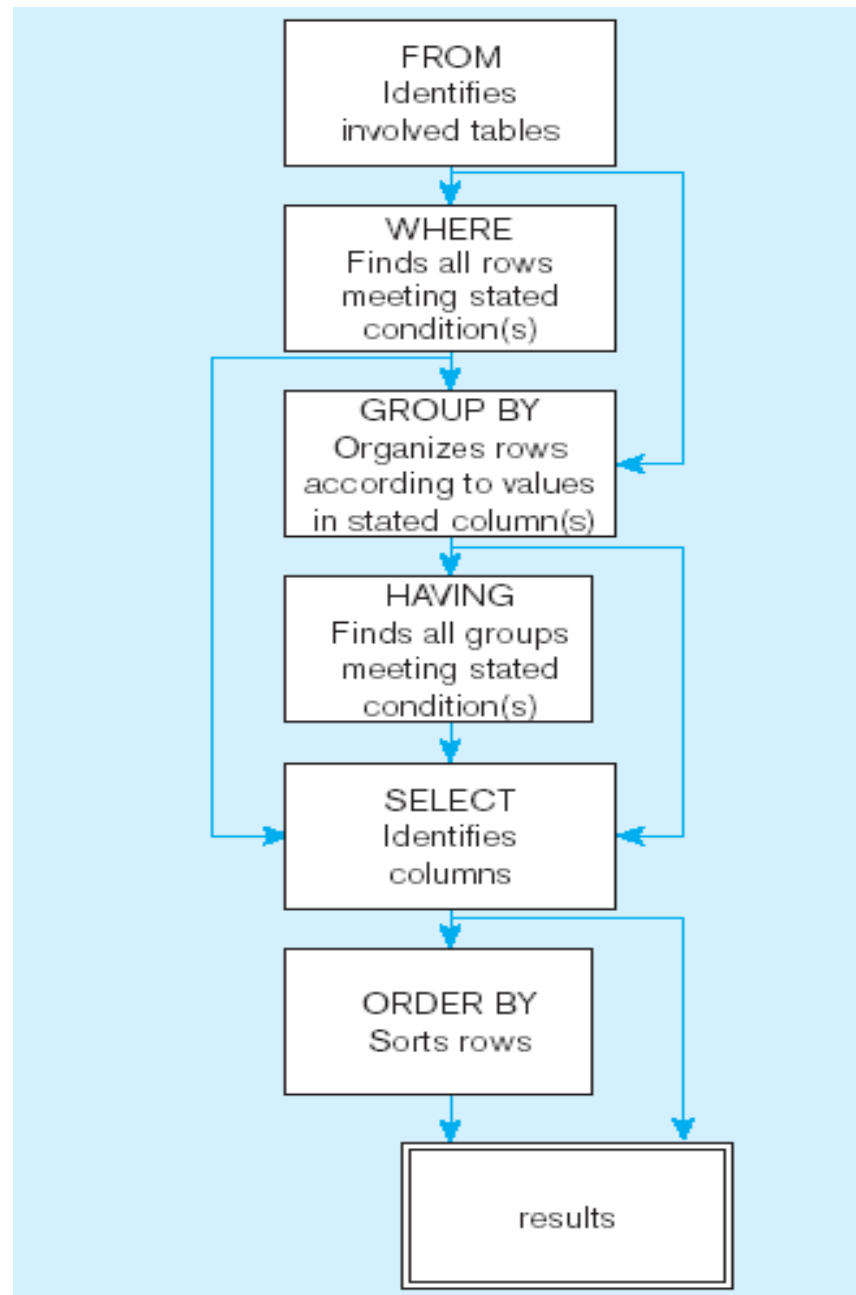
Update statement

- `UPDATE Customers`
 `SET ContactName = 'Alfred Schmidt',`
 `City= 'Frankfurt'`
 `WHERE CustomerID = 1;`
- Multi records
- `UPDATE Customers`
 `SET ContactName='Juan'`
 `WHERE Country='Mexico';`

SELECT Statement

- Used for queries on single or multiple tables
- Clauses of the SELECT statement:
 - **SELECT**
 - List the columns (and expressions) that should be returned from the query
 - **FROM**
 - Indicate the table(s) or view(s) from which data will be obtained
 - **WHERE**
 - Indicate the conditions under which a row will be included in the result
 - **GROUP BY**
 - Indicate categorization of results
 - **HAVING**
 - Indicate the conditions under which a category (group) will be included
 - **ORDER BY**
 - Sorts the result according to specified criteria

Figure 7-10
SQL statement
processing
order (adapted
from van der
Lans, p.100)



SELECT Example

- Find products with standard price less than \$275

```
SELECT PRODUCT_NAME, STANDARD_PRICE  
FROM PRODUCT_V  
WHERE STANDARD_PRICE < 275;
```

Table 7-3 Comparison Operators in SQL

<i>Operator</i>	<i>Meaning</i>
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
!=	Not equal to

Table 7-3: Comparison Operators in SQL

SELECT Example Using Alias

- Alias is an alternative column or table name

```
SELECT CUST.CUSTOMER AS NAME,  
       CUST.CUSTOMER_ADDRESS  
FROM CUSTOMER_V CUST  
WHERE NAME = 'Home Furnishings';
```

SELECT Example Using a Function

- Using the COUNT ***aggregate function*** to find totals

```
SELECT COUNT(*) FROM ORDER_LINE_V  
WHERE ORDER_ID = 1004;
```

Note: with aggregate functions you can't have single-valued columns included in the SELECT clause

Min and Max

- `SELECT MIN(column_name)
FROM table_name
WHERE condition;`
- `SELECT MAX(column_name)
FROM table_name
WHERE condition;`

Top

- `SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;`
- The following SQL statement selects the first three records from the "Customers" table
- `SELECT TOP 3 * FROM Customers;`
- `SELECT * FROM Customers
LIMIT 3;`

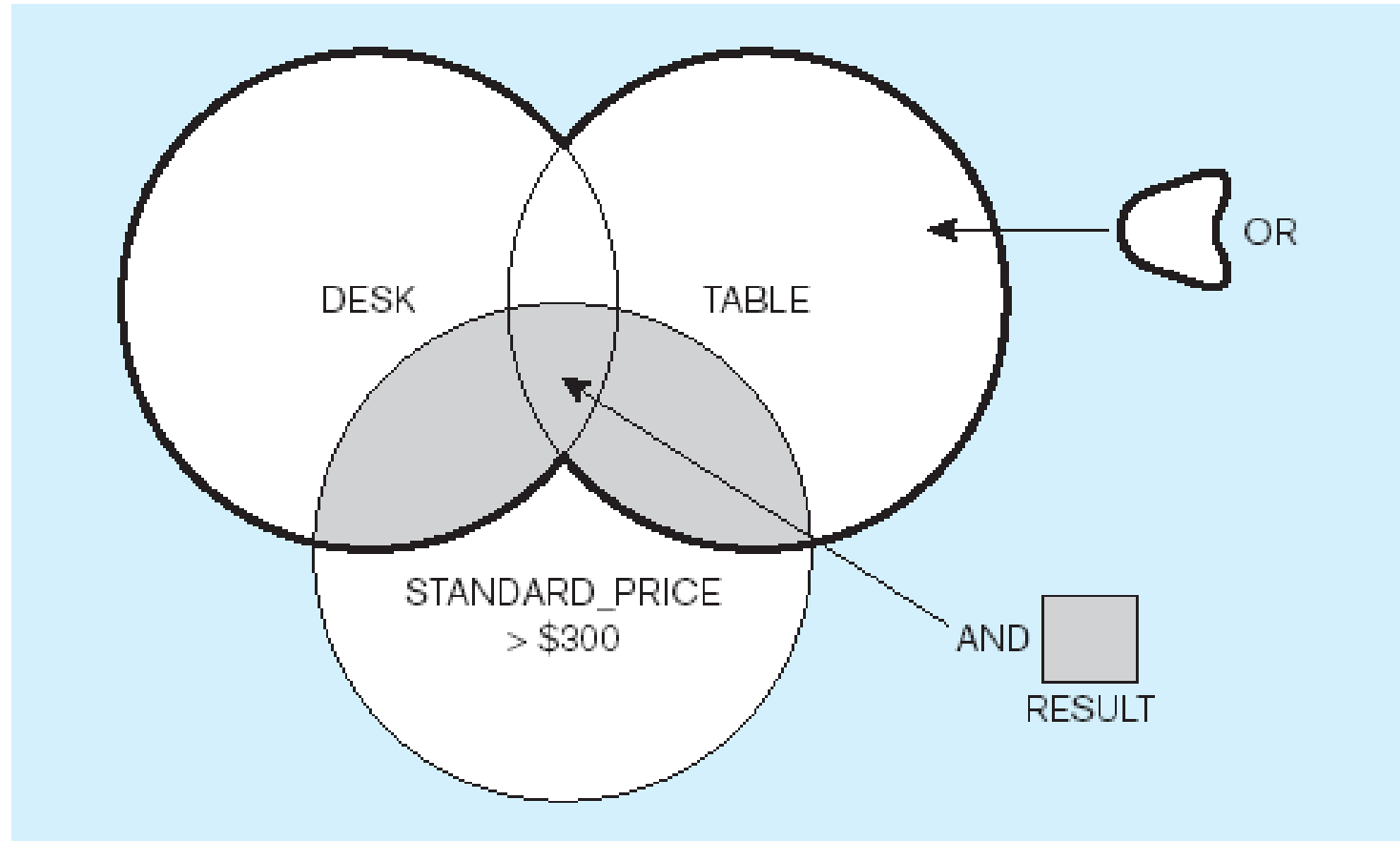
SELECT Example–Boolean Operators

- **AND**, **OR**, and **NOT** Operators for customizing conditions in WHERE clause

```
SELECT PRODUCT_DESCRIPTION, PRODUCT_FINISH,  
       STANDARD_PRICE  
FROM PRODUCT_V  
WHERE (PRODUCT_DESCRIPTION LIKE '%Desk'  
       OR PRODUCT_DESCRIPTION LIKE '%Table')  
       AND UNIT_PRICE > 300;
```

Note: the LIKE operator allows you to compare strings using wildcards. For example, the % wildcard in '%Desk' indicates that all strings that have any number of characters preceding the word "Desk" will be allowed

Venn Diagram from Previous Query



SELECT Example – Sorting Results with the ORDER BY Clause

- Sort the results first by STATE, and within a state by CUSTOMER_NAME

```
SELECT CUSTOMER_NAME, CITY, STATE  
FROM CUSTOMER_V  
WHERE STATE IN ('FL', 'TX', 'CA', 'HI')  
ORDER BY STATE, CUSTOMER_NAME;
```

Note: the IN operator in this example allows you to include rows whose STATE value is either FL, TX, CA, or HI. It is more efficient than separate OR conditions

SELECT Example—

Categorizing Results Using the GROUP BY Clause

- For use with aggregate functions
 - **Scalar aggregate**: single value returned from SQL query with aggregate function
 - **Vector aggregate**: multiple values returned from SQL query with aggregate function **(via GROUP BY)**

```
SELECT CUSTOMER_STATE, COUNT(CUSTOMER_STATE)
FROM CUSTOMER_V
GROUP BY CUSTOMER_STATE;
```

Note: you can use single-value fields with aggregate functions if they are included in the GROUP BY clause

SELECT Example—

Qualifying Results by Categories
Using the HAVING Clause

- For use with GROUP BY

```
SELECT CUSTOMER_STATE, COUNT(CUSTOMER_STATE)
FROM CUSTOMER_V
GROUP BY CUSTOMER_STATE
HAVING COUNT(CUSTOMER_STATE) > 1;
```

Like a WHERE clause, but it operates on groups (categories), not on individual rows. Here, only those groups with total numbers greater than 1 will be included in final result

Update Statement

- Modifies data in existing rows
- **UPDATE PRODUCT_T SET UNIT_PRICE = 775 WHERE PRODUCT_ID = 7;**