### **MySQL Workbench Tutorial**

#### CONTENTS

2	MySC	L Workbench. Introduction
	2.1.1	Considering that you have just finished the installation, now the GUI will open
	(MyS	QL Workbench) with the configuration already set. (Figure 1)
	2.1.2	Now you are connected to the MySQL database server and you can start
	creati	ng a database. Select the "Schemas" tab to view a list of databases running on the
	locals	server. (Figure 4)
	2.1.3	There will be some example schemas installed. You can expand the schema and
	the "٦	Tables" section to view the tables in the database (Figure 5)
	2.1.4	Right click on a table name (e.g. city) and click on "Select rows – limit 1000". This
	will sł	now you a list of the first 1000 rows in the table, using a simple SELECT SQL query
	that y	ou can view in the top window. (Figure 6)6
	2.1.5	Right click on a table name (e.g. city) and click on "Alter Table". This allows you
		ew and modify the table definition (table name, columns, constraints, indexes,
	foreig	n keys, etc.). (Figure 7)6
3	Creat	ing a Database with MySQL Workbench7
3	.1 C	reating the database schema and tables (DDL)8
	3.1.1	Creating the database schema8
	3.1.2	Creating the database tables8
4	Addin	g data into a database with MySQL Workbench14
4	.1 U	sing the GUI14
4	.2 R	unning an SQL query that contains the data15
5	Expor	ting data from a table16
6	Writir	ng SQL queries16

#### 2 MYSQL WORKBENCH. INTRODUCTION

This section presents a tutorial for working with MySQL Workbench: connecting to the database server, working with schemas, tables and data with a visual interface/GUI.

## 2.1.1 Considering that you have just finished the installation, now the GUI will open (MySQL Workbench) with the configuration already set. (Figure 1)

This will be your visual tool to connect to the database server, used for working with databases and tables, viewing and adding data and writing SQL queries. Click on "Local Instance MySQL80" in the "MySQL Connections" list to open a connection to the database server.

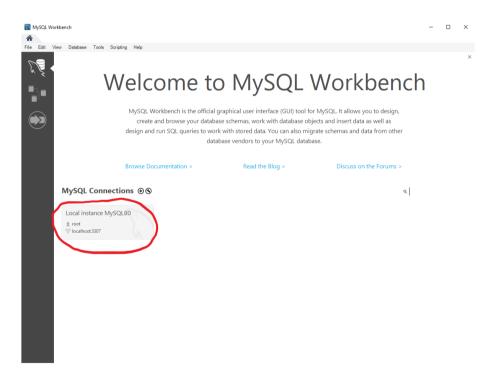


Figure 1 Welcome to MySQL Workbench

**Note: if you don't see the connection in the list**, then you might have skipped a step in the installation guide. It's no problem, you can create a new connection by clicking on the "+"

icon next to "MySQL Connections" as in (Figure 2). Then click "Ok" and move on to the next step.

🕅 Setup New Conn	ection		- 🗆 X			
Connection Name:	Local instance MySQL		Type a name for the connection			
Connection Method:	Standard (TCP/IP)	$\stackrel{\scriptstyle\checkmark}{\sim}$ Method to use to connect to the RDBMS				
Parameters SSL	Advanced					
Hostname:	127.0.0.1	Port: 3306	Name or IP address of the server host - and TCP/IP port.			
Username:	root		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 blank to select it later.			
Configure Server	Management		Test Connection Cancel OK			

Figure 2 Setup New Connection

Connect to the MySQL server using the root password provided during the setup (e.g. ewis2020). (Figure 3)

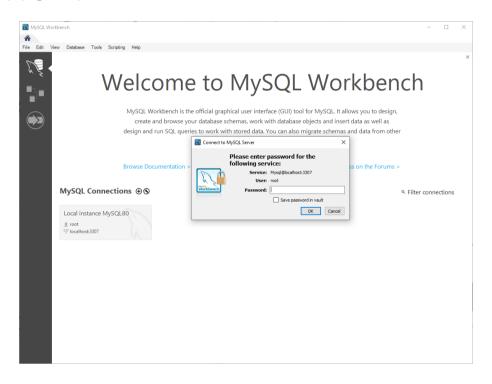


Figure 3 Connect using root password

2.1.2 Now you are connected to the MySQL database server and you can start creating a database. Select the "Schemas" tab to view a list of databases running on the local server. (Figure 4)

Image: Construction MySQL80 ×         File Ecit View Query Databases Server Tools Scripting Help         Image: Construction Scripting Help         Image: Constructing Help         Image	
∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑         ∑	
Navigstor         Query 1 ×         SQLAdditions           SCHEAAS         ●         ■         ■         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●	
SCHEMAS ♥ 🗀 🗟   🖉 🖉 🔕   ◎ 💿   ◎ ◎ 🗑   Limit to 1000 rows <	
A liter objects 1 Automatic context help is disabled. Use the toolbar to manually get help	6
<ul> <li>▶ Sakila Autoinatic context help is disabled, use the tooliar to inalitating get help to be availabled. Use the tooliar to inalitating get help to be availabled. Use the context help is disabled, use the context help is disabled. Use the context help is disabled, use the context help is disabled. Use the context help is disabled, use the context help is disabled. Use the context help is disabled, use the context help is disabled. Use thelp is dis disabled. Use thelp is disabled. Use the context he</li></ul>	, 101
Administrating Schemas c Context Help Snippets	
Information Output	
and management of the second sec	
No object selected # Time Action Duration, Message Duration,	Fetch
Object Info Session	

Figure 4 MySQL Workbench. Schemas

## 2.1.3 There will be some example schemas installed. You can expand the schema and the "Tables" section to view the tables in the database (Figure 5)

Note: The example schemas will be shown if you followed all steps in the setup tutorial. If not, then you may skip this part and keep it as reference for when you will create your own schemas and tables.

🕅 MySQL Workbench
A Local instance MySQL80 ×
File Edit View Query Database S
Navigator
SCHEMAS 🚸
Q Filter objects
<ul> <li>sys</li> <li>world</li> <li>Tables</li> <li>city</li> <li>country</li> <li>countrylanguage</li> <li>Views</li> <li>Stored Procedures</li> <li>Functions</li> </ul>
Administration Schemas
Information Out

Figure 5 Expanding database schema and tables

2.1.4 Right click on a table name (e.g. city) and click on "Select rows – limit 1000". This will show you a list of the first 1000 rows in the table, using a simple SELECT SQL query that you can view in the top window. (Figure 6)

MySQL Workbench												- 11 >
Local instance MyS	SQL80 ×											
ile Edit View Query	y Databa	se Serve	er Tools S	cripting Help								
8 5 6 8			ö 💭									Ø <b>–</b> –
		_										
Navigator		city								SQLAdditions		
CHEMAS	•			A C L		Limit to 1000 row	• 🎭 🕑 🔍 👖 🖻			< ▶   []	ያ 📆 i Jump	p to 👻
Filter objects		1	SELECT *	FROM world.c:	ity;					Autom	atic cont	ext help is
sakila												e toolbar to
sys 🗐 world						Query Ta	b					elp for the
🔻 👘 Tables												sition or to
city												atic help.
<ul> <li>country</li> <li>countrylangu</li> </ul>	1308											
Views	age											
Stored Procedure	es	<							>			
Functions												
		Result G	Srid 🔢 📣			dit: 💋 🔜 🗄	🛛 Export/Import: 🏭 📷   Wrap Cell Content: 🔢	Fetch rows:				
		ID	Name	CountryCod		reputation			Result Grid			
		1	Kabul	AFG	Kabol	1780000			Grid			
		2	Qandahar Herat	AFG AFG	Qandahar Herat	237500 186800						
		4	Mazar-e-Sha		Balkh	127800	Result Grid Tab					
		5	Amsterdam	NLD	Noord-Holland		Result Grid Tab		Form Editor			
		6	Rotterdam	NLD	Zuid-Holland	593321						
			Haag	NLD	Zuid-Holland	440900						
		8	Utrecht	NLD	Utrecht Noord-Brahan	234323			$\sim$			
dministration Schemas		city 1 $\times$		NITT	NINAPA-Brahan	1044		Apply	Revert	Context Help	Snippets	
C		Output ::										
nformation		output										
	^		ion Output	*								
Table: city	^			tion			Message					Duration / Fetch
nformation Table: city Columns:	^	_∏ Acti ≠	Time Ac	* tion LECT * FROM work	l.city LIMIT 0, 100	0	Message 1000 row(s) returned					Duration / Fetch 0.016 sec / 0.000 s
Table: city Columns:		Acti	Time Ac 1 11:09:20 SE				-					
Table: city Columns: ID AI PK	I K	<ul> <li>Acti</li> <li>#</li> <li>1</li> <li>2</li> </ul>	Time Ac 1 11:09:20 SE 2 11:09:23 SE	LECT * FROM work	.country LIMIT 0,	1000	1000 row(s) returned					0.016 sec / 0.000 s
Table: city Columns: ID AI Name ch	I K har(:	<ul> <li>Acti</li> <li>#</li> <li>1</li> <li>2</li> <li>3</li> </ul>	Time         Ac           1         11:09:20         SE           2         11:09:23         SE           3         11:09:25         SE	LECT * FROM work	.country LIMIT 0, .countrylanguage	1000 LIMIT 0, 1000	1000 row(s) returned 239 row(s) returned					0.016 sec / 0.000 s 0.015 sec / 0.000 s
Table: city Columns: ID AI PR Name ch CountryCode ch	I K har( har( har(	Acti     #     0     2     0     4	Time         Ac           1         11:09:20         SE           2         11:09:23         SE           3         11:09:25         SE           4         11:09:29         SE	LECT * FROM work LECT * FROM work LECT * FROM work	.country LIMIT 0, .countrylanguage .country LIMIT 0,	1000 LIMIT 0, 1000 1000	1000 row(s) returned 239 row(s) returned 984 row(s) returned					0.016 sec / 0.000 s 0.015 sec / 0.000 s 0.016 sec / 0.000 s

Figure 6 Select rows – limit 1000

# 2.1.5 Right click on a table name (e.g. city) and click on "Alter Table". This allows you to view and modify the table definition (table name, columns, constraints, indexes, foreign keys, etc.). (Figure 7)

Note: On small screens (e.g. laptop) some tabs will not show properly. If this is the case, you may drag the bottom tab (Action Output) to fill less space, so that you can view the (more important) column definitions.

1 월 1 🖗 1 🕹 🗿 🛃 4 Nigator	on i li i i i i i i i i i i i i i i i i i												SQLAdditions	۵	
HEMAS ®	Table Na					st	ema: world					\$	< ⊳   <u>Te</u> 1	anp to	
Piter objects	Charset/Colat			✓ latin1_bin			ine: InnoDB						Automat	ic context he	eln is
sakila sys world Tables	Comments:			. autom		·) eng	ne: Inue					0	disabled. manually	Use the tool get help for aret position	bar t r the
► Country/inquise View Stored Procedures Princtions	10     Name     CountryCode     District     Population	INT CHAR(IS) CHAR(3) CHAR(20) INT			Default/Expression										
	Column Name: Charset/Colation:							Data Type: Default:		⊖ treat					
mation	Charset/Collation:	Views						✓ Default:	O Wrtuel	Stored Not Null Unsigned Generated					
rmation	Charset/Collabon:	ieys Triggers Pert	tioning Options					✓ Default:	Virtual Primary Key Binary	Not Null	🗌 Zero Fil				
mation able: city olumns: m int Al	Charset/Collation: Commenta:	feys Trippers Part	Honing Options					✓ Default:	Virtual Primary Key Dinary	Not Null	Zero Fil		Contac Help Sr	ippets	
nation ble: city blumns: ID int AI PK	Charset/Calatons Commenta: Columna Indexes Foreign I Culput	ieys Trippers Pert	Moning Options					✓ Default:	Virtual Primary Key Dinary	Not Null	🗌 Zero Fil		Contact Help St	ippets	
able: city olumns: ID PK Name char(35) CountryCode char(35)	Columns Indexes Foreign 1 Columns Indexes Fo	-					1000ge 2000 späil si dured	✓ Default:	Virtual Primary Key Dinary	Not Null	🗌 Zero Fil		Context Help Si	lopets Duration 0/016 sec	
able: city columns: 10 int A1 Pic (har(35)) CountyCode (char(3)) Date: (char(3))	Columne Indexes Foreign 1 Culput Culput The Action Action	• ROM world.cky LIMIT (	L 1000					✓ Default:	Virtual Primary Key Dinary	Not Null	🗌 Zero Fil		Contect Help Sr	Duration	c / 0.00
mation able: city blumns: ID PK Name char(35) CountyCode char(30) Detret char(30)	Colume Indexes Foreign1 Column Indexes Foreign1 Column Indexes Foreign1 Column Index Attain T Trans Attain T Tr	<ul> <li>ROM world city LIMIT ( ROM world country LIMIT)</li> </ul>	L 1000 IT 0, 1000	×			1000 row(s) returned	✓ Default:	Virtual Primary Key Dinary	Not Null	🗌 Zero Fil		Contex Help Sr	Duration , 0.016 sec	c / 0.000 c / 0.000
Table: city Colums: ID PK Name char(35) CountryCode char(35) Datest char(32)	Columns Indexes Foreign1 Columns Indexes Foreign1 Column Indexes Foreign1 Column Intervention Di Actor Odyot Thma Actor T	<ul> <li>RDM world.cty LIMIT 0</li> <li>RDM world.country LIM</li> <li>RDM world.countryleng</li> </ul>	L 1000 IT 0, 1000 wage LIMIT 0, 1000				1000 row(s) returned 239 row(s) returned	✓ Default:	Virtual Primary Key Dinary	Not Null	🗌 Zero Fil		Context Help Sr	Duration 0.016 sec 0.015 sec	c / 0.000 c / 0.000 c / 0.000
Name char(35) CountryCode char(3) District char(20)	Colume Indexes Foreign 1 Column Indexes Foreign 1 Column Indexes Foreign 1 Column Index Anter  1 The Anter 1 The Anter  1 The Anter 1 T	<ul> <li>ROM world city LIMIT ( ROM world country LIM ROM world countryleng ROM world country LIM</li> </ul>	L 1000 ITT 0, 1000 Logge LIMIT 0, 1000 ITT 0, 1000				1000 row(s) returned 239 row(s) returned 584 row(s) returned	✓ Default:	Virtual Primary Key Dinary	Not Null	🗌 Zero Fil		Contex Help Sr	Duration 0.016 sec 0.015 sec 0.016 sec	c / 0.000 c / 0.000 c / 0.000 c / 0.000

Figure 7 Alter Table

#### **3** CREATING A DATABASE WITH MYSQL WORKBENCH

This section describes how to create a simple database shown in Figure 8 using MySQL Workbench, then adding data and writing simple queries. While this example is quite simple, the tutorial is valid for creating and working with any MySQL database in MySQL Workbench.

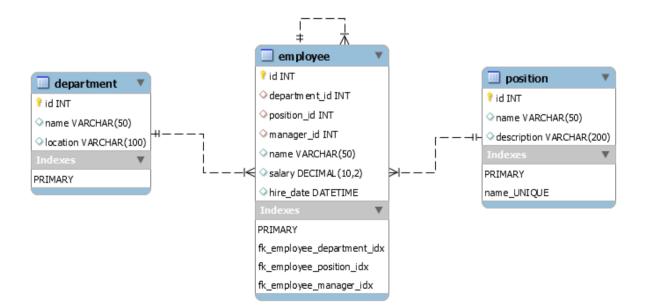


Figure 8 Example schema. Company database

#### 3.1 Creating the database schema and tables (DDL)

The database schema and tables in this tutorial are created according to the representation in Figure 8. It is common to define the schema first and then to implement it into a DBMS. Now, this design step is already done for the simple example, and we are just implementing the schema into MySQL using MySQL Workbench.

#### 3.1.1 Creating the database schema

Select the "Schemas" tab and right click in the blank area under the "SCHEMAS" list view. Click on "Create Schema". A new database is about to be created. Change the "Name" field to *company* then click "Apply". A popup window will open. Click "Apply" and then "Finish".

MySQL Workbench			– n ×
Local instance MySQL80 ×			- Ľ ^
File Edit View Query Databa	ise Server Tools Scripting Help		
8 8 6 6 6 6 6	1 <b>6 1</b>		Ø 🔲
Navigator	company - Schema 🛛 🗙		SQLAdditions
SCHEMAS 🚸	Name: company	Specify the name of the schema here. You can use any combination of ANSI letters,	◄ ▷   ፲ ? <sup>4</sup> / <sub>1</sub> ?   Jump to
Filter objects	Rename References	Refactor model, changing all references found in view, triggers, stored procedures and functions from the old schema name to the new one.	Automatic context help is
▶ Sakila ▶ Sys ▶ World	Charset/Collation: Default Charset 🗸 Default Collatioi 🗸	The character set and its collation selected here will be used when no other	disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Administration Schemas	Schema		
		Apply Revert	Context Help Snippets
Schema: world	Output		
	Action Output •		
	# Time Action 1 17:33:02 SELECT * FROM world.countrylanguage LIMIT	Message 0, 1000 984 row(s) returned	Duration / Fetch 0.015 sec / 0.000 se
	2 17:33:05 SELECT * FROM world.country LIMIT 0, 1000	239 row(s) returned	0.000 sec / 0.000 se
	3 17:33:11 SELECT * FROM world.city LIMIT 0, 1000	1000 row(s) returned	0.015 sec / 0.000 sec
Object Info Session			

Figure 9 Creating a new database schema

#### 3.1.2 Creating the database tables

Now that the database is created, we need to add the tables: *department, employee, position*.

Click on the arrow next to the *company* schema, then right click on "Tables", and click on "Create Table". A new table is about to be created. Change the "Table Name" field to *department*.

#### 3.1.2.1 Adding columns and constraints

To add a new column to a table, click on the first blank row under "Column Name". Then, change the name to *id* to define the first attribute for the *department* table<sup>1</sup>. The *id* attribute will be of type INT (leave INT in the "Datatype" field) and will be the PRIMARY KEY for the *department* table. Select AI (Auto Increment) constraint to make the id auto generated.

Then, continue to add the other column definitions as shown in Figure 10. After adding all columns, click "Apply". A popup window will open, where you can view the DDL command generated by MySQL Workbench:

CREATE TABLE `company`.`department` ( `id` INT NOT NULL AUTO\_INCREMENT, `name` VARCHAR(50) NULL, `location` VARCHAR(100) NULL, PRIMARY KEY (`id`));

Click on "Apply" to execute the DDL command and create the table into the database, then click on "Finish".

<sup>&</sup>lt;sup>1</sup> The *id* column makes it easier to uniquely identify records in the created tables, otherwise the PRIMARY KEY can also be a combination of attributes (in this case *name* and *location* can uniquely identify a department)

MySQL Workbench		– 0 ×
Local instance MySQL80 ×		
File Edit View Query Database	e Server Tools Scripting Help	
		Ø <b>–</b> –
Navigator	department - Table ×	SQLAdditions
SCHEMAS (%	Table Name: department Schema: company	<ul> <li>Image: Image: Ima</li></ul>
Q Filter objects		
v 📄 company	Charset/Collation: Default Charset V Default Collation V Engine: InnoDB V	Automatic context help is disabled. Use the toolbar to
骨 Tables 智 Views 骨 Stored Procedures 骨 Functions	Comments:	manually get help for the current caret position or to toggle automatic help.
▶ 📄 sakila	Column Name Datatype PK NN UQ B UN ZF AI G Default/Expression	
▶ Sys ▼ Sworld	INT         INT <td></td>	
🔻 🛅 Tables		
▶ city		
<ul> <li>▶ country</li> <li>▶ countrylanguage</li> <li>♥ Views</li> <li>♥ Stored Procedures</li> <li>♥ Functions</li> </ul>		
	Column Name: Id Data Type: INT	
	Charset/Collation: Default Charset   Default Collation  Default:	
	Comments: Storage: Virtual Stored	
	🗹 Primary Key 🔽 Not Null 🗌 Unique	
Administration Schemas	Binary Unsigned Zero Fil	
Information		
	Auto Increment Generated	
Table: city	Columns Indexes Foreign Keys Triggers Partitioning Options	
Columns:	Apply Reve	rt Context Help Snippets
ID AI PK	Output	Concarney Compyco
Name char(	Action Output	
CountryCode char( District char(	# Time Action Message	Duration / Fetch
Population int	17 14:06:12 DROP DATABASE 'new_schema'     0 row(s) affected	0.063 sec
Object Info Session		0.000 / 0.000

Figure 10 *department* table definition

3.1.2.2 Then, proceed to add the other tables in the same way: employee, position.

🔝 MySQL Workbench		- 0 ×
Local instance MySQL80 ×		
File Edit View Query Databas	e Server Tools Scripting Help	
		Ø <b>–</b> –
Navigator	employee - Table 🗙	SQLAdditions
SCHEMAS 🚸	Table Name: employee Schema: company	< ▶ 🕼 🕺 Jump to -
<ul> <li>Filter objects</li> </ul>	Table Name: employee Schema: company	-
▼ 🗟 company ▼ 🛱 Tables	Charset/Colation: utf8mb4 v utf8mb4_0900_ai_c v Engine: InnoD8 v	Automatic context help is disabled. Use the toolbar to
department     employee	Comments:	manually get help for the current caret position or to
position	>	toggle automatic help.
Views	Column Name Datatype PK NN UQ B UN ZF AI G Default/Expression	coggie duconduce neipi
Stored Procedures	P id         INT         IV         I         IV         IV           ◊ department_id         INT         IV	
sakila		
sys	♦ manager id INT NULL	
► 🗐 world	◇ name VARCHAR(50)	
	◇ salary DECIMAL(10,2)	
	hire_date     DATETIME     DATETIME	
	Column Name: manager_id Data Type: INT	
	Charset/Collation: Default Charset V Default Collation V Default: NULL	
	Comments: Storage: Virtual Stored	
	Primary Key Not Null Unique	
Administration Schemas	Binary Unsigned Zero Fill	
Information	Auto Increment Generated	
^		
Table: employee	Columns Indexes Foreign Keys Triggers Partitioning Options	
Columns: id int AI P	Apply Revert	Context Help Snippets
department_id int		
position_id int manager id int	Output	
name varchar	Action Output	
salarv decimal *	≠ Time Action Message	Duration / Fetch
Object Info Session	<ul> <li>27 15:43:18 Apply changes to department</li> <li>Changes applied</li> </ul>	
a allocation and a second		

Figure 11 *employee* table definition

MySQL Workbench			– o ×
Local instance MySQL80 ×			
File Edit View Query Databas	Server Tools Scripting Help		
			Ø <b></b> _
Navigator	osition - Table $\times$		SQLAdditions
SCHEMAS 🚸	Table Name: position	Schema: company	◄ ▶   12 1/2 Jump to
Q Filter objects	Table Name: position	Schema: company	
v 🗟 company	Charset/Collation: utf8mb4 v utf8mb4_0900	_ai_c ~ Engine: InnoDB ~	Automatic context help is
Tables		^	disabled. Use the toolbar to
department     employee	Comments:		manually get help for the
position		×	current caret position or to toggle automatic help.
Tiews	Column Name Datatype PK NN UQ	B UN ZF AI G Default/Expression	toggie automatic neip.
Stored Procedures Functions	Id         INT         ✓         □                ∩ name         VARCHAR(50)         □         ✓                √ description         VARCHAR(200)         □         □		
<ul> <li>B sakila</li> </ul>	◇ name VARCHAR(50)		
▶  sys	description     VARCHAR(200)		
▶ 🗎 world			
	Column Name:	Data Type:	
	Charset/Collation:	Default:	
	Comments:	Storage: Virtual O Stored	
		Primary Key Not Null Unique	
Administration Schemas		Binary Unsigned Zero Fill	
Information			
Information		Auto Increment Generated	
Table: position			
Table: position	olumns Indexes ForeignKeys Triggers Partitioning Options		
Columns: id int AI PK		Apply Revert	Context Help Snippets
name varchar(50)			
description varchar(200)			
	Action Output •		
	# Time Action	Message	Duration / Fetch
Object Info Session	27 15:43:18 Apply changes to department	Changes applied	
object mo bession		~ F1	

Figure 12 *position* table definition

After creating the tables, they will show in the schemas navigator as shown in Figure 13.

Navigator
SCHEMAS 🚸
<b>Q</b> Filter objects
<ul> <li>Company</li> <li>Tables</li> <li>department</li> <li>employee</li> <li>position</li> <li>Views</li> <li>Stored Procedures</li> <li>Functions</li> <li>sakila</li> <li>sys</li> <li>world</li> </ul>
Administration Schemas

Figure 13 The tables are now visible in schemas navigator

#### 3.1.2.3 *Defining the relationships between tables*

After the tables are created, we can add all the relationships as defined in the example schema (Figure 8). For this, we need to modify the existing tables by right clicking on a table name and then clicking on "Alter Table".

#### **Department – Employee**

#### Relationship description

- A department can have one or more employees
- An employee can belong to a single department
- Implementation: one-to-many relationship

#### Implementation

We need to define a FOREIGN KEY in the *employee* table to reference the *department* table. Right click on *employee* table, then click on "Alter Table". Switch to "Foreign Keys" tab and create the FOREIGN KEY constraint as shown in Figure 14. You have to define a name, the referenced table, the column that will be set as foreign key, the referenced column and the ON UPDATE and ON DELETE rules (we will use CASCADE for all foreign keys in this example).

File: Exclaired: View: Query Database: Server: Tods: Sorging: Help File: Exclaired: Query Database: Server: Tods: Sorging: Help Sorging: Company: C	🕅 MySQL Workbench			– 0 ×
Administration       Scheme         Image: employee       Context fielp         Image: employee       Context fielp <td>Local instance MySQL80 ×</td> <td></td> <td></td> <td></td>	Local instance MySQL80 ×			
Nungace     SURMAS     Stream	File Edit View Query Database	e Server Tools Scripting Help		
Sciences       Image: Company         Image: Comments:       Image: Comments:         Image: Comment:       Im		<b>6 1</b>		Ø <b>[</b> ]
Administration       Schemas         Imployee       Outcom/fech         Columns:       Columns:         Imployee       Outcom/fech	Navigator	employee - Table $~\times~$		SQLAdditions
Administration Schemes		Table Name- employee	Schema: company	🐟 ◄ ▶   🜆 💏   Jump to 🔹
Administration       Schemas         Information       Intermet department department department de department department de department department de department depa				* *
Administration       Schemas         Information       Columns         Information	V Company	Charset/Collation: utf8mb4 V u	tf8mb4_0900_ai_c ~ Engine: InnoDB ~	
Wrevs Fractions       Foreign Key Name       Referenced Table         Referenced Table       Foreign Key Name         Columns:       Gastion Object         Gepartment, Id       It         Referenced Table       Name         Columns:       Apply<	department     employee	Comments:		manually get help for the current caret position or to
Bord Procedures			Earcian You Ontions	toggle automatic help.
Prections     Prections <td>The Stored Procedures</td> <td></td> <td>Column Referenced Column</td> <td>~</td>	The Stored Procedures		Column Referenced Column	~
Administration       Schemas         Information       Information         Columns:       Iddition         dipartment_id       nt         Administration       Schemas         Information       Columns:         Information       Columns:         Information       Columns:         Information       Columns:         Information       Int AIP         Output       Output         Columns:       Int AIP         Information       Missage         Duput       Output         Output       Int AIP         Information       Missage         Duput       Duput         Output       Duput         Information       Missage         Duput       Time         Action       Missage         Duput       Duput		fk_employee_manager `company`.`employee`	department_idid	
Administration Schemas Information Columns: d department_d rane Columns: d d department_d rane Columns: d d d d d d d d d d d d d d d d d d d	sys	fk_employee_position `company`.`position`		·
Administration Schemas  Information  Table: employee Columns:     int AIP     postion / d     nt     manage_id     vt charge     Columns:     department_d     nt     manage_id     vt charge     columns     department_d     manage_id     vt charge     columns     department_d     vt charge     columns     vt charge     columns     context Help Snippets     context Help Snippet	► 🗐 world			
Administration Schemas  Information  Administration Schemas  Information  Columns:  Int AIP  position id  it  Administration id  it  Apply Revert Context Help Snippets  Output  Apply Revert Context Help Snippets  Duration / Fetch  Z 154.318 Apply changes to department Changes applied			salary	
Administration Schemas Information Table: employee Columns: Indexs Foreign Keys ogers Partitioning Options Columns: Index Foreign Keys ogers Partitioning Columns: Index Foreign Keys oge			hire_date	
Table: employee Columns: Index Foreign Keys ogers Partitioning Options Columns: Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Output Columns Index Foreign Keys ogers Partitioning Options Columns Index			- Foreign Key Comment	
Table: employee Columns: Index Foreign Keys ogers Partitioning Options Columns: Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Output Columns Index Foreign Keys ogers Partitioning Options Columns Index			•	^
Table: employee Columns: Index Foreign Keys ogers Partitioning Options Columns: Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Output Columns Index Foreign Keys ogers Partitioning Options Columns Index				
Table: employee Columns: Index Foreign Keys ogers Partitioning Options Columns: Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Output Columns Index Foreign Keys ogers Partitioning Options Columns Index				
Table: employee Columns: Index Foreign Keys ogers Partitioning Options Columns: Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Columns Index Foreign Keys ogers Partitioning Options Output Columns Index Foreign Keys ogers Partitioning Options Columns Index				
Table: employee     Columns:     int AIP       Output     Output       Dutput     Output       Output     Output	Administration Schemas			
Table: employee     Columns:     int AIP       Output     Output       Dutput     Output       Output     Output	Information			
Columns:     Int Al P       didupartment_id     nt       namage_id     nt       name     victure       iditiv     int       int		<pre></pre>		
Columns: id department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid ship department_id nt managerid of 27 154318 Apply hanges to department changes appled	Table: employee			
id     int AI P       object Info     Section       id     int AI P       opsition_id     int       manager_id     int       id     int </td <td>Columns:</td> <td>to agrine to agrine to agrine to the to any</td> <td></td> <td></td>	Columns:	to agrine to agrine to agrine to the to any		
position_id     int       manager /d     int       name     varchar       sharv     der main       4     Tma       Action     Message       Dilate life (     Section       0     27       154.318     Apply changes to department       Changes appled	id int AI P		Apply Reve	Context Help Snippets
name varchar varchar in a Action output and the same varchar in the Action of the Acti	position id int	Output		
Chiert Info. Session Other Tofo. Session Other	name varchar	Action Output •		
Object Info Session 27 15:43:18 Apply changes to department Changes applied	salarv decimal *		-	Duration / Fetch
		27 15:43:18 Apply changes to department	Changes applied	

Figure 14 Adding foreign keys to the *employee* table. Department - Employee

#### **Employee – Position**

#### Relationship description

- An employee can have a single position in the company
- There can be more employees with the same position in the company (A position may be assigned to one or more employees)
- Implementation: one-to-many relationship

#### Implementation

We need to define a FOREIGN KEY in the *employee* table to reference the *position* table. In the "Foreign Keys" tab in *employee* table, add the FOREIGN KEY constraint as shown in Figure 15.

MySQL Workbench									– 0 ×
File Edit View Query Databas		elp							Ø .
Navigator	employee - Table $~\times~$								SQLAdditions
SCHEMAS (%)	Table Name:	employee		Schema:	company			\$	◄ ▶   🛐 🌴   Jump to ▼
Company	Charset/Collation:	utf8mb4 $\vee$	utf8mb4_0900_ai_c $ \smallsetminus $	Engine:	InnoDB		~		Automatic context help is
Tables     department     employee     position	Comments:						< >		disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Views	Foreign Key Name Refe	enced Table	Column	Reference	Column	Foreign Key Optic	ins		toggie automatic help.
Stored Procedures Functions		pany`.`department`	🗌 id			On Update:	CASCADE	~	
▶ 📑 sakila ▶ 📑 sys		pany`.`employee` pany`.`position`	☐ department_id ☑ position_id ☐ manager_id	id		On Delete:	CASCADE	~	
😫 world			name salary hire_date			Skip	in SQL generatio	'n	
						- Foreign Key Com	ment		
								^	
dministration Schemas									
nformation									
Table: employee	Columns Indexes Foreign Key	> Triggers Partitioning	< Options		>	L		~	
Columns: id int AI P department_id int	Output							evert	Context Help Snippets
position_id int manager_id int	Action Output								
name varchan salarv decimal ¥	# Time Action				Message				Duration / Fetch
bject Info Session	<ul> <li>27 15:43:18 Apply changes</li> <li>20 15 43 53 Apply changes</li> </ul>				Changes ap	plied			

Figure 15 Adding foreign keys to the *employee* table. Employee - Position

#### **Employee – Employee**

#### Relationship description

- An employee can have a single manager (which is also an employee of the company)
- There can be more employees that have the same manager (A manager may be assigned to one or more employees)

• Implementation: one-to-many/self-referencing relationship

#### Implementation

We need to define a FOREIGN KEY in the *employee* table to reference the *employee* table. In the "Foreign Keys" tab in *employee* table, add the FOREIGN KEY constraint as shown in Figure 16.

_											
MySQL Workbench										- 0	×
Local instance MySQL80 ×											
	se Server Tools Scripting	lelp									
8 8 6 8 8 8 8	i 🕡 🦊									0	
Navigator	employee - Table ×								SOLAdditions		
SCHEMAS ®		employee							< ► I? ¶o	Jump to	-
Q. Filter objects	Table Name						$\approx$	~			
Company	Charset/Collation	utf8mb4 $\vee$	utf8mb4 $\sim$ utf8mb4_0900_ai_c $\sim$		InnoDB	~			Automatic co		
Tables							~		disabled. Us		
department     employee	Comments:								current care	et help for t	
position							$\vee$			tomatic help	
Views Stored Procedures	Foreign Key Name Refe	renced Table	Column	Reference	d Column	Foreign Key Optio	ons		coggie du		
Functions		pany`.`department`	id			On Update:	CASCADE	$\sim$			
🕨 🗐 sakila		pany`,`employee` pany`,`position`	department_id position id			On Delete:	CASCADE	~			
· 📄 sys · 🗎 world			manager id	id							
0 1010			name salarv			Skip	in SQL generatio	n			
			hire_date								
						- Foreign Key Com	ment	_			
								^			
Administration Schemas											
nformation											
^	<	>	<		>			$\sim$			
Table: employee	Columns Indexes Foreign Key	s Triggers Partitionin	g Options								
Columns:						A	pply Re	evert	Context Help Snipp	ate	
id int AIP department id int											
position id int	Output										
name varchar	Action Output •										
salarv decimal *	# Time Action				Message					Duration / Fetch	^
Object Info Session	<ul> <li>27 15:43:18 Apply changes</li> <li>20 15 43:53 h d d</li> </ul>				Changes a	pplied					
		1									

Figure 16 Adding foreign keys to the *employee* table. Employee - Employee

#### 4 ADDING DATA INTO A DATABASE WITH MYSQL WORKBENCH

This section shows how you can add/update/remove data to existing tables using MySQL Workbench.

#### 4.1 Using the GUI

To add data, Right click on a table and click on "Select Rows – Limit 1000". A view will open as in Figure 17 where you can view the existing data and you can add or remove data. To add a new row, click on an empty row and edit the fields. You can add multiple rows at once. After adding the rows to the table, you have to click on "Apply" to save them into the database. A popup window will open with the SQL commands that will be applied. Click on "Apply" and then "Finish". To remove an existing row, Right click on the button to the left side and click on "Delete Row(s)". As before, you have to click on "Apply" to remove the rows from the database. Alternatively, you can click on "Revert" and the operation will be cancelled (the rows will be restored to the table view).

Result Grid	•	Filter Rows:	E	idit: 🔏 🗮	<b>=</b>	Export/Import:	Wrap Cell Co	ntent: 🚹	
id	name	location							
1	finance	New York							
2	PR	Paris							
	marketing	Geneva							
NULL	NULL	NULL							
epartment	1 ×								Apply
								<u> </u>	

Figure 17 Adding rows to a table

#### 4.2 Running an SQL query that contains the data

To open a new query tab in Workbench, click on "File" then click on "New Query Tab". A blank tab will open where you can write your SQL query. There is one more step before running the query. You have to select the target database. Double click on the *company* database. The name will turn bold. Now you can apply the query by clicking on the yellow bolt icon. See Figure 18.

A Local instance MySQL80 ×	×	- 0 ×
File Edit View Query Datab	abase Server Tools Scripting Help	
	📾 🛺 🙀 🚽 🗾 execute query	Ø <b>D</b>
Navigator	SQL File 5" ×	SQLAdditions
SCHEMAS (%)	Image: A contract of the second se	<ul> <li>▶   10 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2</li></ul>
<ul> <li>C Filter objects</li> <li>C company</li> <li>C Tables</li> <li>F department</li> <li>P opsition</li> <li>Wess</li> <li>Sored Procedures</li> <li>Sored Procedures</li> <li>Sakia</li> <li>Sakia</li> <li>Sakia</li> </ul>	1 • INSERT INTO 'employee' VALUES (1,1,1,NULL,'Big Hoss',10000.00,'2018-10-18 00:00'0)',(2,1,1,1,'Harc Rick select database (double click)	Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Material Science		Context Help Snippets
Administration Schemas		Context Help Snippets
Administration Schemas	Cutput	Context Help Snippets
Information	Cutput	
	Cutput	Duration / Fetch
Information	<ul> <li>Output</li> <li>Time Action</li> <li>Message</li> </ul>	Duration / Fetch
Information	Culput     Culput	Duration / Fetch At Do used by doubl
Information	Output     Output     Concernent of the selected Select the default     O 1 1553.01 show create table company department     I 2 1553.51 show create table company department     I row(s) returned	It DB to be used by doubl 0.000 sec 0.000 sec
Information	Output     Output     Time Action     Action     1 15:5:42 inhow create table department     2 115:53:51 show create table department     2 215:53:51 show create table company department     1 row(s) returned     3 16:03:15 Apply changes to company2     Changes appled	It DB to be used by doubl 0.000 sec 0.000 sec
Information	Culput     Culput     Action Output     Time Action     If Time A	t DB to be used by doubl 0.000 sec 0.000 sec 10.000 sec 0.000 sec 0.0000 sec 0.000 sec 0.0000 sec 0.0000 sec 0.000 sec 0.0000 sec 0.0

Figure 18 MySQL Workbench – SQL Editor

#### 5 EXPORTING DATA FROM A TABLE

To save the result of a query, you can export as *csv* file as shown in Figure 19. Give the file a name and then open the file with a text editor. You will see the content in *csv* format that you can use later to process the data.

<							~		
Re	id 1	name	Filter Rows:	Edit: 🔏 🚦	•	Export/Impor	i i wr	ap Cell Content:	IA [
	2	PR	Paris						
./	NULL	marketing NULL	Geneva						
		1 ×							Apply

Figure 19 Exporting data to a *csv* file



Figure 20 Data exported as csv file

#### **6 WRITING SQL QUERIES**

This section presents some SQL query examples on the *company* database. To write an SQL query in MySQL Workbench, see Figure 18.

#### Q1

```
-- Write a query in SQL to display all the information of the employees -- SELECT * FROM employee;
```

#### Q2

-- List the employees name and salary. Show only those having the salary between 5000 and 7000 EUR and order them descending by salary --SELECT name, salary FROM employee WHERE salary BETWEEN 5000 AND 7000 ORDER BY salary DESC;

#### Q3

-- List the employees name and the name of their department --SELECT emp.name AS employee, dept.name AS department FROM employee AS emp INNER JOIN department AS dept ON dept.id = emp.department\_id;

#### **Q**4

-- Get the number of employees in the company --SELECT COUNT(id) AS employee\_count FROM employee;

#### Q5

```
-- Get the number of employees from department id 4 --
SELECT department_id, COUNT(id) AS employee_count FROM employee
WHERE department_id = 4
GROUP BY department id;
```

#### **Q**6

-- Get the number of employees for each department --SELECT department\_id, COUNT(id) AS employee\_count FROM employee GROUP BY department\_id;

#### Q7

```
-- Get the average salary for each department --
SELECT department_id, AVG(salary) AS average_salary FROM employee
GROUP BY department_id;
```

#### Q8

```
-- Get the number of analysts from department id 3 --
SELECT department_id, position_id, COUNT(id) AS employee_count FROM employee
WHERE department_id = 3 AND position_id = 3;
```

#### **Q9**

```
-- Get the number of analysts for each department --
SELECT department_id, position_id, COUNT(id) AS employee_count FROM employee
WHERE position_id = 3
GROUP BY department_id;
```

#### Q10

```
-- List the employees name and the name of their manager --
SELECT manager.name AS manager_name, emp.name AS employee_name FROM employee
AS emp
INNER JOIN employee AS manager ON emp.manager_id = manager.id
ORDER BY manager_name ASC;
```

#### Q11

```
-- List the employees name, salary, the name of their position --
SELECT emp.name AS employee, pos.name AS POSITION, emp.salary FROM employee
AS emp
INNER JOIN POSITION AS pos
ON pos.id = emp.position id;
```

#### Q12

```
-- List the employees name, salary, the name of their department and position
--
SELECT emp.name AS employee, dept.name AS department, pos.name AS POSITION,
emp.salary FROM employee AS emp
INNER JOIN department AS dept
ON dept.id = emp.department_id
INNER JOIN POSITION AS pos
ON pos.id = emp.position_id;
```