

# MYSQL QUERIES

Objective:

Install Mysql workbench on windows 10

Create Database and run queries on Mysql workbench.

Step 1:

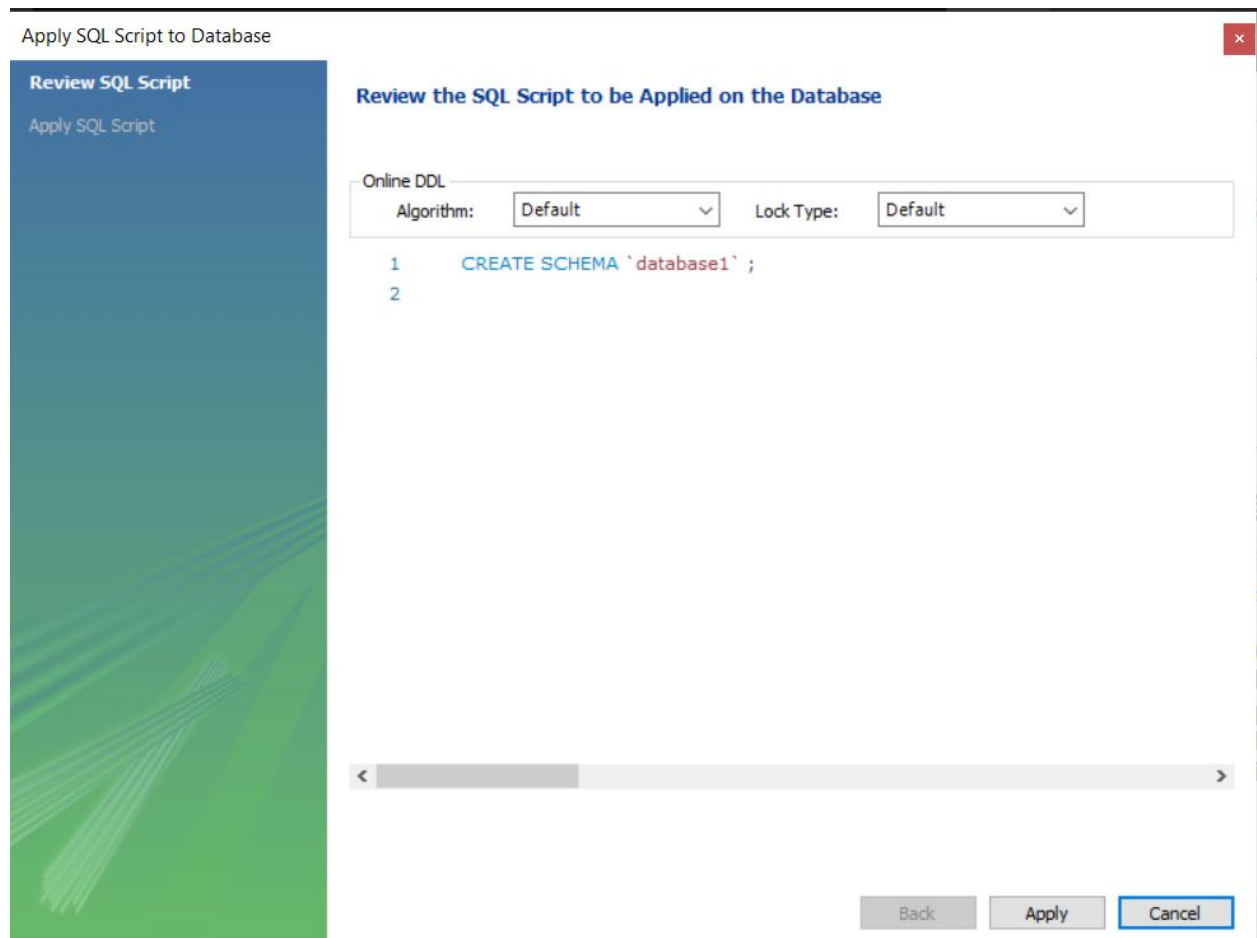
Download and install Mysql workbench 8.0 from [mysql.com](https://dev.mysql.com/downloads/workbench/)

Step 2:

From start open Mysql workbench after installation is complete and create a new database 'database1' and click on apply

Step 3:

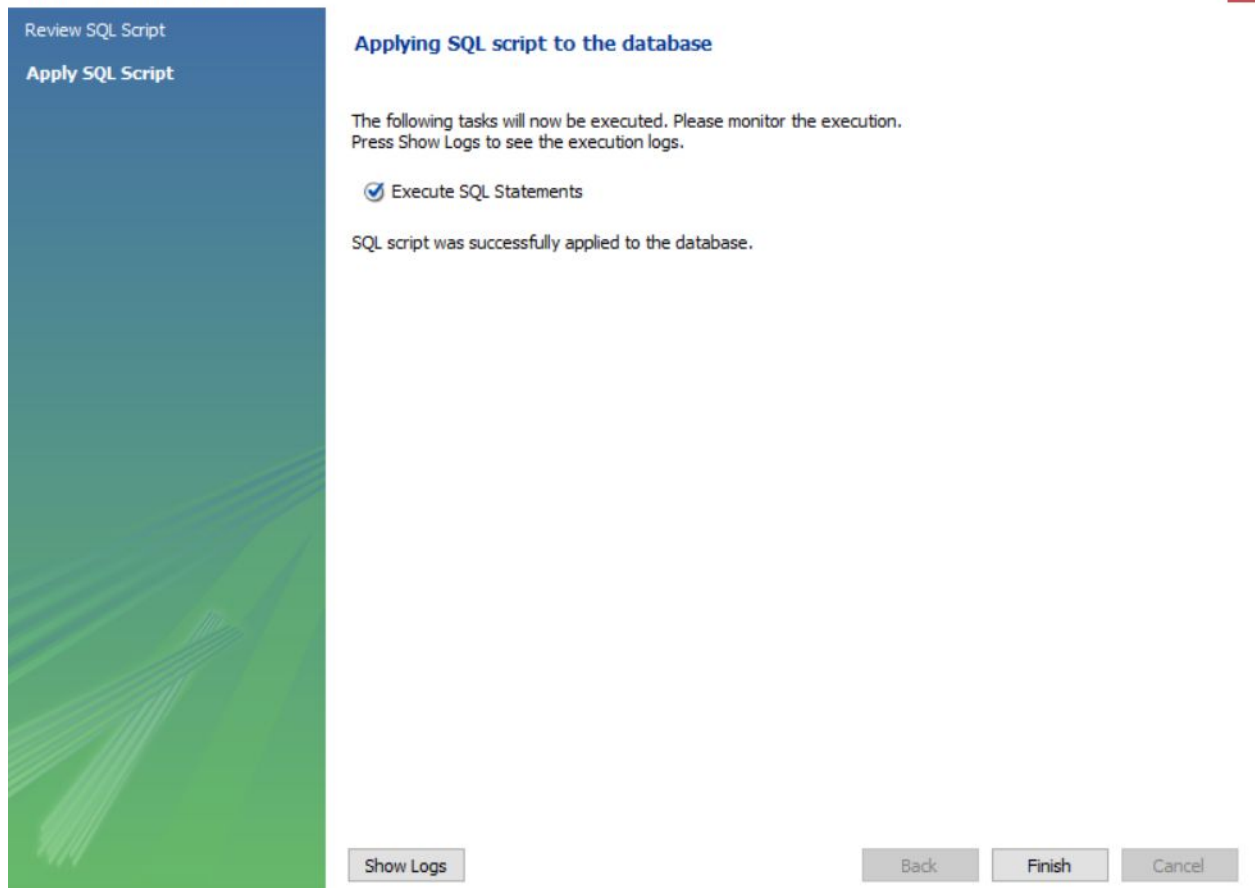
A new window appears select 'Apply'




Step 3:

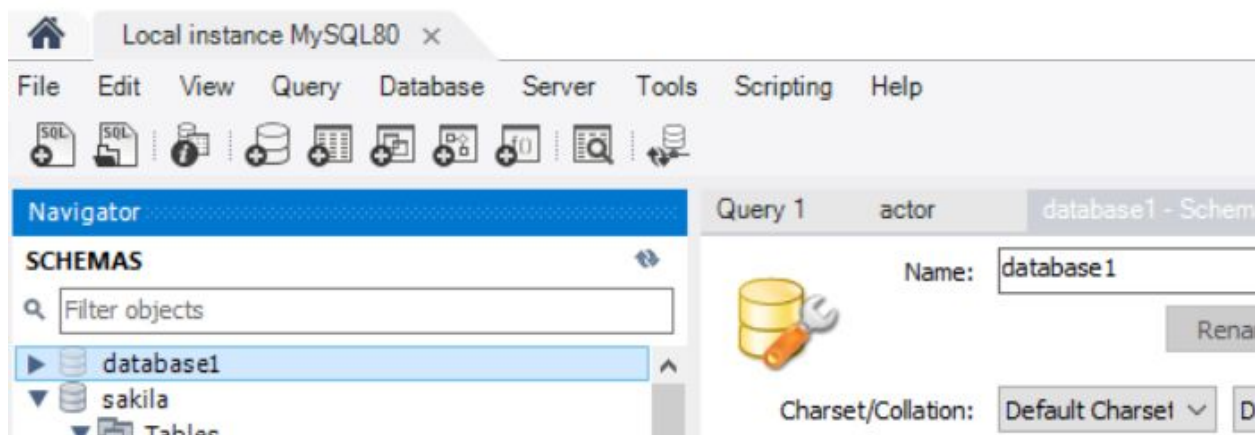
Select 'Finish'

Apply SQL Script to Database



Step 4:  
Create database1.

 MySQL Workbench



Step 5: add employee table and columns: employee id, first name, last name and Age.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Query 1 actor city database3 - Schema **employee table - Table**

Table Name:  Schema: **database1**

Charset/Collation:   Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
employee id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
first name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
last name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Age	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:  Data Type:

Charset/Collation:

Default:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

## Review SQL Script

Apply SQL Script

## Review the SQL Script to be Applied on the Database

Online DDL

Algorithm:

Default

Lock Type:

Default

```
1 CREATE TABLE `database1`.`employee table` (  
2   `employee id` INT NOT NULL,  
3   `first name` VARCHAR(45) NOT NULL,  
4   `last name` VARCHAR(45) NULL,  
5   `Age` INT NOT NULL);  
6
```

<

>

Back

Apply

Cancel

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

database1

Tables

employee table

Views

Stored Procedures

Functions

database3

Tables

Views

Stored Procedures

Functions

sakila

Tables

Views

Stored Procedures

Functions

store

sys

world

Administration Schemas

Schema: database1

Query 1 actor city database3 - Schema employee table - Table x

Table Name: employee table Schema: database1

Charset/Collation: utf8mb4 utf8mb4\_0900\_ai Engine: InnoDB

Comments:

Column Name: Age Data Type: INT(11)

Charset/Collation: Default Charset Default Collation Default:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	17:50:56	Apply changes to database3	Changes applied	
3	17:51:28	Apply changes to database3	No changes detected	
4	17:53:40	Apply changes to employee table	Error 1064: You have an error in your SQL syntax; ...	
5	17:57:29	Apply changes to employee table	Changes applied	

## Step 7: Add data to the columns.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: database1 employeeetable actor database1.employeeetable employeeetable - Table employeeetable - Table employeeetable employeeetable x

SCHEMAS

Filter objects

database1

Tables

employeeetable

Columns

- employeeid
- first name
- last name
- Age

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

database3

Tables

Views

Stored Procedures

Functions

sakila

Tables

actor

Administration Schemas

Information

Table: employeeetable

Columns:

- employeeid int(11) AI PK
- first name varchar(45)
- last name varchar(45)
- Age int(11)

1 • SELECT \* FROM database1.employeeetable;

2

Result Grid

employee id	first name	last name	Age
101	malligai	malli	22
102	bri	man	23
103	mun	bun	25
104	gandhi	gandhi	25
105	mahatma	mahatma	26
106	indira	gandhi	27
107	jawaharlal	nehru	28
108	lakshmi	lakshmi	29
109	pillayar	ganesha	30
110	sivoham	sivoham	22
111	buddhi	buddhi	23
112	siddhi	siddhi	24
113	mahaveer	mahaveer	25
114	hanuman	hanuman	26
115	chalisa	chalisa	27
116	sai	sai	28
117	baba	baba	29
118	auro	auro	30
119	mira	mira	31
120	saraswathi	saraswathi	32
121	vishnu	vishnu	28

employeeetable 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
54	20:36:48	SELECT * FROM database1.employeeetable LIMIT 0, 1000	24 row(s) returned	0.000 sec / 0.000 sec



## Step 8 : start querying database - ORDER BY query

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'database1' expanded, showing 'employeeetable' and 'actor' tables. The 'employeeetable' table structure is shown with columns: employeeid, first name, last name, and Age. The main query editor contains the following SQL query:

```
1 • SELECT * FROM database1.employeeetable ORDER BY Age;
2
```

The 'Result Grid' displays the query results, sorted by Age. The data is as follows:

employee id	first name	last name	Age
101	malligai	mali	22
110	sivoham	sivoham	22
102	bri	man	23
111	buddhi	buddhi	23
112	siddhi	siddhi	24
103	mun	bun	25
104	gandhi	gandhi	25
113	mahaveer	mahaveer	25
105	mahatma	mahatma	26
114	hanuman	hanuman	26
106	indira	gandhi	27
115	chalisa	chalisa	27
107	jawaharlal	nehru	28
116	sai	sai	28
121	vishnu	vishnu	28
108	lakshmi	lakshmi	29
117	baba	baba	29
122	brahma	brahma	29

The bottom panel shows the 'Output' tab with the following log entries:

#	Time	Action	Message	Duration / Fetch
54	20:36:48	SELECT * FROM database1.employeeetable LIMIT 0, 1000	24 row(s) returned	0.000 sec / 0.000 sec
55	20:39:03	SELECT * FROM database1.employeeetable LIMIT 0, 1000	24 row(s) returned	0.000 sec / 0.000 sec
56	20:43:56	SELECT * FROM database1.employeeetable ORDER BY Age LIMIT ...	24 row(s) returned	0.000 sec / 0.000 sec

## Step 9 : start querying database - OR query

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: database1.employeeetable actor database1.employeeetable employeeetable - Table employeeetable - Table employeeetable

SCHEMAS

Filter objects

database1

Tables

employeeetable

Columns

- employeeid
- first name
- last name
- Age

Indexes

Foreign Keys

Administration Schemas

Information

Table: employeeetable

Columns:

- employeeid int(11) AI PK
- first name varchar(4)
- last name varchar(4)
- Age int(11)

Object Info Session

Limit to 1000 rows

1 • SELECT \* FROM database1.employeeetable WHERE Age = '22' OR Age = '30';

2

3

Result Grid

employee id	first name	last name	Age
101	malligai	malli	22
109	pillayar	ganesha	30
110	sivoham	sivoham	22
118	auro	auro	30
123	murari	mu	30
NULL	NULL	NULL	NULL

employeeetable 8 x

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
59	20:55:25	SELECT * FROM database1.employeeetable WHERE Age = 2...	5 row(s) returned	0.000 sec / 0.000 sec
60	20:57:15	SELECT * FROM database1.employeeetable WHERE Age = 2...	5 row(s) returned	0.000 sec / 0.000 sec

## Step 10 : start querying database - AND query

The screenshot shows the MySQL Workbench interface for a local instance of MySQL 8.0. The left sidebar displays the 'SCHEMAS' tree with 'database1' expanded, showing tables like 'employeeetable'. The main editor window contains a SQL query: `SELECT * FROM database1.employeeetable WHERE Age = '22' AND Age = '30';`. The 'AND' operator is highlighted in yellow. Below the query editor, the 'Result Grid' shows a table with columns 'employee id', 'first name', 'last name', and 'Age', all containing 'NULL'. The bottom panel shows the 'Output' tab with an 'Action Output' table.

#	Time	Action	Message	Duration / Fetch
60	20:57:15	SELECT * FROM database1.employeeetable WHERE Age = '22'...	5 row(s) returned	0.000 sec / 0.000 sec
61	20:58:33	SELECT * FROM database1.employeeetable WHERE Age = '22' AND Age = '30';	0 row(s) returned	0.000 sec / 0.000 sec

## Step 11: start querying database - NOT query

The screenshot shows the MySQL Workbench interface with a query executed. The query is `SELECT * FROM database1.employeeetable WHERE NOT Age = '22';`. The result grid displays 22 rows of employee data. The output pane shows the execution details for two queries, both returning 22 rows.

**Table: employeeetable**

**Columns:**

- employee id: int(11) AI PK
- first name: varchar(4)
- last name: varchar(4)
- Age: int(11)

**Result Grid:**

employee id	first name	last name	Age
114	hanuman	hanuman	26
115	chalisa	chalisa	27
116	sai	sai	28
117	baba	baba	29
118	auro	auro	30
119	mira	mira	31
120	saraswathi	saraswathi	32
121	vishnu	vishnu	28
122	brahma	brahma	29
123	murari	mu	30
1241	meera	mer	31

**Output:**

#	Time	Action	Message	Duration / Fetch
62	21:00:59	SELECT * FROM database1.employeeetable WHERE NOT Age = '22'...	3 row(s) returned	0.000 sec / 0.000 sec
63	21:01:13	SELECT * FROM database1.employeeetable WHERE NOT Age = '22' LI...	22 row(s) returned	0.000 sec / 0.000 sec

## Step 12: start querying database - AND, OR query

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • SELECT * FROM database1.employeeetable WHERE Age = '22' AND (Age = '30' OR Age = '23') ;
2
3
```

The left sidebar shows the Schemas pane with the following structure:

- database1
  - Tables
    - employeeetable
      - Columns
        - employee id
        - first name
        - last name
        - Age
      - Indexes
      - Foreign Keys

The bottom left pane shows the Information tab for the **Table: employeeetable** with the following columns:

Columns:	Details
employee id	int(11)
id	AI PK
first name	varchar(4)
last name	varchar(4)
Age	int(11)

The right pane shows the Result Grid with the following data:

employee id	first name	last name	Age
NULL	NULL	NULL	NULL

The bottom right pane shows the Output tab with the following data:

#	Time	Action	Message	Duration / Fetch
62	21:00:59	SELECT * FROM database1.employeeetable WHERE NOT Age = '22' AND Age = '30' ...	3 row(s) returned	0.000 sec / 0.000 sec
63	21:01:13	SELECT * FROM database1.employeeetable WHERE NOT Age = '22' LIMIT 0, 1000	22 row(s) returned	0.000 sec / 0.000 sec
64	21:05:53	SELECT * FROM database1.employeeetable WHERE Age = '22' AND (Age = '30' OR ...	0 row(s) returned	0.016 sec / 0.000 sec

## Step 13 : start querying database - ORDER BY ASC query

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

database1

Tables

employee table

Columns

employee id

first name

last name

Age

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

database3

Tables

Views

Stored Procedures

Administration Schemas

Information

Table: employee table

Columns:

employee id int(11)  
AI PK  
first name varchar(4)  
last name varchar(4)  
Age int(11)

Object Info Session

database1.employee table actor database1.employee table employee table - Table employee table - Table employee table sql 1

Limit to 1000 rows

1 • SELECT \* FROM database1.employee table ORDER BY Age ASC;

2

3

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

employee id	first name	last name	Age
101	malligai	mali	22
110	sivoham	sivoham	22
102	bri	man	23
111	buddhi	buddhi	23
112	siddhi	siddhi	24
103	mun	bun	25
104	gandhi	gandhi	25
113	mahaveer	mahaveer	25
105	mahatma	mahatma	26
114	hanuman	hanuman	26
106	indira	gandhi	27
115	chalisa	chalisa	27
107	jawaharlal	nehru	28
116	rai	rai	28

employee table 13

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
41	13:30:15	SELECT * FROM database1.employee table LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
42	13:31:37	SELECT * FROM database1.employee table LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec

## Step 14 : start querying database - ORDER BY DESC query

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: database1.employeeetable actor database1.employeeetable employeeetable - Table employeeetable - Table employeeetable sql1\*

SCHEMAS

Filter objects

database1

Tables

employeeetable

Columns

- employee id
- first name
- last name
- Age

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

database3

Tables

Views

Stored Procedures

Administration Schemas

Information

Table: employeeetable

Columns:

- employee int(11)
- id AI PK
- first name varchar(4)
- last name varchar(4)
- Age int(11)

Object Info Session

1 • SELECT \* FROM database1.employeeetable ORDER BY Age DESC;

2

3

Result Grid

Filter Rows:

Limit to 1000 rows

employee id	first name	last name	Age
120	saraswathi	saraswathi	32
119	mira	mira	31
1241	meera	mer	31
109	pillayar	ganesha	30
118	auro	auro	30
123	murari	mu	30
108	lakshmi	lakshmi	29
117	baba	baba	29
122	brahma	brahma	29
107	jawaharlal	nehru	28
116	sai	sai	28
121	vishnu	vishnu	28
106	indira	gandhi	27
115	chalisa	chalisa	27

employeeetable 15 x

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
67	21:21:17	SELECT * FROM database1.employeeetable ORDER BY Age DESC LIMIT 0, 1000	24 row(s) returned	0.000 sec / 0.000 sec
68	21:21:34	SELECT * FROM database1.employeeetable ORDER BY Age DESC LIMIT 0, 1000	24 row(s) returned	0.000 sec / 0.000 sec



## Step 15 : start querying database - LIKE PATTERN '%2%' query

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: loyeetable actor database1.employeeetable employeeetable - Table employee table employeeetable - Table employeeetable sql11

**SCHEMAS**

Filter objects

- database1
  - Tables
    - employeeetable
      - Columns
        - employee id
        - first name
        - last name
        - Age
      - Indexes
      - Foreign Keys
      - Triggers
    - Views
    - Stored Procedures
    - Functions
  - database3
    - Tables
    - Views
    - Stored Procedures
    - Functions

Administration Schemas

Information

**Table: employeeetable**

**Columns:**

- employee id int(11) AI PK
- first name varchar(4)
- last name varchar(4)
- Age int(11)

Object Info Session

1 • SELECT \* FROM database1.employeeetable WHERE Age LIKE '%2%';

2

3

Result Grid

employee id	first name	last name	Age
101	malligai	mali	22
102	bri	man	23
103	mun	bun	25
104	gandhi	gandhi	25
105	mahatma	mahatma	26
106	indira	gandhi	27
107	jawaharlal	nehru	28
108	lakshmi	lakshmi	29
110	sivoham	sivoham	22
111	buddhi	buddhi	23
112	siddhi	siddhi	24
113	mahaveer	mahaveer	25
114	hanuman	hanuman	26
115	chalisa	chalisa	27
116	sai	sai	28
117	baba	baba	29

employeeetable 16 x

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
66	21:18:00	SELECT * FROM database1.employeeetable ORDER BY A...	24 row(s) returned	0.000 sec / 0.000 sec



Step 16: start querying database - LIKE PATTERN '%2' query

The screenshot shows the MySQL Workbench interface with a query executed in the SQL editor. The query is: `SELECT * FROM database1.employeeetable WHERE Age LIKE '%2';`

The result grid displays the following data:

employee id	first name	last name	Age
101	malligai	mali	22
110	sivoham	sivoham	22
120	saraswathi	saraswathi	32
NULL	NULL	NULL	NULL

The bottom panel shows the Action Output with the following entries:

#	Time	Action	Message	Duration / Fetch
66	21:18:00	SELECT * FROM database1.employeeetable ORDER BY Ag...	24 row(s) returned	0.000 sec / 0.000 sec
67	21:21:17	SELECT * FROM database1.employeeetable ORDER BY Ag...	24 row(s) returned	0.000 sec / 0.000 sec

## Step 17: start querying database - LIKE PATTERN '3%' query

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'database1' expanded, showing 'employeeetable' and its columns: 'employee id', 'first name', 'last name', and 'Age'. The main editor window contains the SQL query: `SELECT * FROM database1.employeeetable WHERE Age LIKE '3%';`. The 'Result Grid' shows the following data:

employee id	first name	last name	Age
109	pillayar	ganesha	30
118	auro	auro	30
119	mira	mira	31
120	saraswathi	saraswathi	32
123	murari	mu	30
1241	meera	mer	31
NULL	NULL	NULL	NULL

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
82	22:16:40	SELECT * FROM database1.employeeetable WHERE Age L...	0 row(s) returned	0.000 sec / 0.000 sec
83	22:16:56	SELECT * FROM database1.employeeetable WHERE Age L...	6 row(s) returned	0.000 sec / 0.000 sec

## Step 18 : start querying database - BETWEEN, AND query

The screenshot shows the MySQL Workbench interface with a query executed. The query is:

```
SELECT Age FROM database1.employeeetable WHERE (Age BETWEEN '30' AND '32');
```

The result grid shows the following data:

Age
30
30
31
32
30
31

The bottom panel shows the Action Output with the following details:

#	Time	Action	Message	Duration / Fetch
83	22:16:56	SELECT * FROM database1.employeeetable WHERE Age L...	6 row(s) returned	0.000 sec / 0.000 sec
84	14:01:58	SELECT Age FROM database1.employeeetable WHERE (A...	6 row(s) returned	0.000 sec / 0.000 sec

## STEP 19: IS NOT NULL QUERY

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: loyeetable actor database1.employeetable employeetable - Table employee table employeetable - Table employeetable sqj1\*

**SCHEMAS**

Filter objects

database1

Tables

employeetable

Columns

- employee id
- first name
- last name
- Age

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

database3

Tables

Views

Administration Schemas

Information

Table: **employeetable**

Columns:

- employee id int(11)
- first name AI PK varchar(4)
- last name varchar(4)
- Age int(11)

Object Info Session

1 • SELECT \* FROM database1.employeetable WHERE Age IS NOT NULL;

2

3

Limit to 1000 rows

Result Grid

employee id	first name	last name	Age
101	malligai	malli	22
102	bri	man	23
103	mun	bun	25
104	gandhi	gandhi	25
105	mahatma	mahatma	26
106	indira	gandhi	27
107	jawaharlal	nehru	28
108	lakshmi	lakshmi	29
109	pillayar	ganesha	30
110	sivoham	sivoham	22
111	buddhi	buddhi	23
112	siddhi	siddhi	24
113	mahaveer	mahaveer	25

employeetable 24 x

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
55	20:39:03	SELECT * FROM database1.employeetable LIMIT 0, 1000	24 row(s) returned	0.000 sec / 0.000 sec
56	20:43:56	SELECT * FROM database1.employeetable ORDER BY Ag...	24 row(s) returned	0.000 sec / 0.000 sec

## STEP 20: IS NULL QUERY

The screenshot shows the MySQL Workbench interface with a query executed in the 'Query' tab. The query is:

```
SELECT * FROM database1.employeeetable WHERE Age IS NULL;
```

The 'Result Grid' shows the following columns: employee id, first name, last name, and Age. All cells in the grid are empty, indicating that no rows were returned by the query.

The 'Information' tab shows the table structure for 'employeeetable':

Columns:	
employee id	int(11)
id	AI PK
first name	varchar(4)
last name	varchar(4)
Age	int(11)

The 'Output' tab shows the execution results:

#	Time	Action	Message	Duration / Fetch
84	14:01:58	SELECT Age FROM database1.employeeetable WHERE (A...	6 row(s) returned	0.000 sec / 0.000 sec
85	14:04:01	SELECT * FROM database1.employeeetable WHERE Age I...	0 row(s) returned	0.000 sec / 0.000 sec

## STEP 21: INSERT QUERY

MySQL Workbench

Mydatabase x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

- database1
  - Tables
    - employeetable
  - Views
  - Stored Procedures
  - Functions
- database3
- information\_schema
- mysql
- performance\_schema
- sakila
- store
- sys
- world

Query 1

employeetable employeetable employeetable x

SQLAdditions

My Snippets

Limit to 1000 rows

```
1 • INSERT INTO database1.employeetable VALUES(125,'raj','raj',25);
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	20:04:06	SELECT * FROM datab...	24 row(s) returned	0.000 sec / 0.000 sec
✗ 2	20:06:20	INSERT INTO databas...	Error Code: 1054. Unknown column 'RAJ' in 'field list'	0.000 sec
✗ 3	20:06:31	INSERT INTO employe...	Error Code: 1054. Unknown column 'RAJ' in 'field list'	0.000 sec
✗ 4	20:08:02	INSERT INTO employe...	Error Code: 1054. Unknown column 'RAJ' in 'field list'	0.000 sec
✗ 5	20:08:51	INSERT INTO employe...	Error Code: 1054. Unknown column 'Raj' in 'field list'	0.000 sec
✓ 6	20:08:58	SELECT * FROM datab...	24 row(s) returned	0.000 sec / 0.000 sec
✓ 7	20:09:28	SELECT * FROM datab...	24 row(s) returned	0.000 sec / 0.000 sec
✗ 8	20:10:40	INSERT INTO databas...	Error Code: 1054. Unknown column 'raj' in 'field list'	0.000 sec
✓ 9	20:12:04	INSERT INTO databas...	1 row(s) affected	0.031 sec

Table: **employeetable**

Columns:

- employee int(11)
- id AI PK
- first name varchar(4)
- last name varchar(4)
- Age int(11)

Object Info Session

## STEP 22: SELECT QUERY TO SEE IF THE INSERT HAPPENED

MySQL Workbench

Mydatabase x

File Edit View Query Database Server Tools Scripting Help

Navigator: Filter objects

SCHEMAS

- database1
  - Tables
    - employeeable
  - Views
  - Stored Procedures
  - Functions
- database3
- information\_schema
- mysql
- performance\_schema
- sakila
- store
- sys
- world

Administration Schemas

Information

Table: employeeable

Columns:

- employee\_id int(11) AI PK
- first\_name varchar(45)
- last\_name varchar(45)
- Age int(11)

Query 1 employeeable employeeable employeeable x

1 • SELECT \* FROM database1.employeeable;

Limit to 1000 rows

Result Grid

employee id	first name	last name	Age
106	indra	gandhi	27
107	jawaharjal	nehru	28
108	lakshmi	lakshmi	29
109	pillayar	ganesha	30
110	sivoham	sivoham	22
111	buddhi	buddhi	23
112	siddhi	siddhi	24
113	mahaveer	mahaveer	25
114	hanuman	hanuman	26
115	chalisa	chalisa	27
116	sai	sai	28
117	baba	baba	29
118	auro	auro	30
119	mira	mira	31
120	saraswathi	saraswathi	32
121	vishnu	vishnu	28
122	brahma	brahma	29
123	murari	mu	30
125	raj	raj	25
1241	meera	mer	31

employeeable4 x

Output

## STEP 23: DESCRIBE OPTION

Query 1 employeeable employeeable employeeable x

1 • DESCRIBE database1.employeeable;

Limit to 1000 rows

Result Grid

Field	Type	Null	Key	Default	Extra
employee id	int(11)	NO	PRI	NULL	auto_increment
first name	varchar(45)	NO		NULL	
last name	varchar(45)	YES		NULL	
Age	int(11)	NO		NULL	