Software Testing Help

☰ Menu

Top 40 Best MySQL Interview Questions And Answers (2019 Questions)

Last Updated: January 14, 2019

List of the Most Popular MySQL Interview Questions with Example Queries (Latest MySQL questions of 2019):

MySQL is an essential part of the modern web application and the knowledge of MySQL is a common requirement for any web developer's job.

A list of basic, as well as advanced level MySQL questions for freshers and experienced professionals, are explained in this article with necessary examples.

These questions will, in turn, help any job seeker to increase their confidence level and face the interview successfully.

Basic Interview Questions	**It Interview Questions and Answers**
MySQL Database	**MySQL Tutorial**

Most Frequently Asked MySQL Interview Questions

Enlisted below are the most popular MySQL interview questions that are commonly asked in the interviews by experts.

Q #1) What is MySQL?

Answer:

MySQL is an open source DBMS which is developed and distributed by Oracle Corporation.

It is supported by most of the popular operating systems, such as Windows, Linux etc. It can be used to develop a different type of applications but it is mainly used for developing web applications.

MySQL uses GPL (GNU General Public License) license so that anyone can download and install it for developing those applications which will be published or distributed freely. But if a user wants to develop any commercial application using MySQL then he/she will need to buy the commercial version of MySQL.

Q #2) What are the features of MySQL?

Answer:

MySQL has several useful features which make it a popular database management software.

Some important features of MySQL are mentioned below.

- It is reliable and easy to use too.
- It is the suitable database software for both large and small application.
- Anyone can install and use it at no cost.
- It is supported by many well-known programming languages, such as PHP, Java, C++, PERL etc.
- It supports standard SQL (Structured Query Language).
- The open source license of MySQL is customizable. Hence, a developer can modify it according to the requirements of the application.

Q #3) What is the default port number of MySQL?

Answer:

The default port number of MySQL is **3306**.

Q #4) How can you find out the version of the installed MySQL?

Answer:

The version of the installed MySQL server can be found out easily by running the following command from the MySQL prompt.

mysql> SHOW VARIABLES LIKE "%version%";

Q #5) What are the advantages and disadvantages of using MySQL?

Answer:

There are several advantages of MySQL which are making it a more popular database system now.

Some significant advantages and disadvantages of MySQL are mentioned below.

Advantages:

- It is well-known for its reliable and secure database management system. Transactional tasks of the website can be done more securely by using this software.
- It supports different types of storage engines to store the data and it works faster for this feature.
- It can handle millions of queries with a high-speed transactional process.
- It supports many advanced level database features, such as multi-level transaction, data integrity, deadlock identification etc.
- Maintenance and debugging process are easier for this software.

Disadvantages:

- It is hard to make MySQL scalable.
- It is not suitable for a very large type of database.
- The uses of stored routine and trigger are limited to MySQL.

Q #6) What is the function of myisamchk?

Answer:

myisamchk is a useful database utility tool that is used to get information about MyISAM database tables.

It is also used for checking, debugging, repairing and optimizing database tables. It is better to use this command when the server is down or when the required tables are not in use by the server.

Syntax:

myisamchk [OPTION] table_name…

The available options of this tool can be retrieved by using the following command.

myisamchk –help

To check or repair all MyISAM tables, the following command will be required for executing from the database directory location.

myisamchk *.MYI

Q #7) What are the purposes of using ENUM and SET data types?

Answer:

ENUM data type is used in the MySQL database table to select any one value from the predefined list.

The value of a particular field can be restricted by defining the predefined list as the field which is declared as ENUM will not accept any value outside the list.

The SET data type is used to select one or more or all values from the predefined list. This data type can also be used to restrict the field for inserting only the predefined list of values like ENUM.

Example:

Run MySQL server from the command prompt and execute the following SQL commands to know the use of ENUM and SET data type.

The following SQL commands create a new database named '**newdb**' and select the database for use.

CREATE DATABASE newdb;

USE newdb;

The following SQL command will create a table named **clients** with the fields ENUM and SET data type.

```sql
1  CREATE TABLE clients (
2      id INT AUTO_INCREMENT PRIMARY KEY,
3      name VARCHAR(50),
4      membership ENUM('Silver', 'Gold', 'Diamond'),
5      interest SET('Movie', 'Music', 'Concert'));
```

Insert query will create two records in the table. ENUM field only accepts data from the defined list.

'**Premium**' value does not exist on the ENUM list. Hence, the value of the ENUM field will be empty for the second record. SET can accept multiple values and both the data will be inserted in the second record.

```
1  INSERT INTO clients (name, membership,interest)
2  VALUES ('Sehnaz','Gold', 'Music'),
3                  ('Sourav','Premium', 'Movie,Concert');
4  SELECT * FROM clients;
```

Q #8) What are the differences between a primary key and foreign key?

Answer:

Database table uses a primary key to identify each row uniquely. It is necessary to declare the primary key on those tables that require to create a relationship among them. One or more fields of a table can be declared as the primary key.

When the primary key of any table is used in another table as the primary key or another field for making a database relation, then it is called a foreign key.

The differences between these two keys are mentioned below.

- The primary key uniquely identifies a record, whereas foreign key refers to the primary key of another table.
- The primary key can never accept a NULL value but foreign key accepts a NULL value.
- When a record is inserted in a table that contains the primary key then it is not necessary to insert the value on the table that contains this primary key field as the foreign key.
- When a record is deleted from the table that contains the primary key then the corresponding record must be deleted from the table containing the foreign key for data consistency. But any record can be deleted from the table that contains a foreign key without deleting a related record of another table.

Example:

Two tables named **manufacturers** and **items** will be created after executing the following two SQL commands.

Here, the primary key of the **manufacturers** table is used as foreign key in the **items** table with the field name **manufacturer_id**. Hence, the **manufacturer_id** field will contain only those values that exist in the **manufacturers** table.

```
1  CREATE TABLE manufacturers (
2      id INT AUTO_INCREMENT PRIMARY KEY,
3      name VARCHAR(50));
4  CREATE TABLE items (
5      id INT AUTO_INCREMENT PRIMARY KEY,
6      name VARCHAR(50),
7      type VARCHAR(50),
8      brand VARCHAR(50),
9      manufacturer_id INT,
10     FOREIGN KEY (manufacturer_id) REFERENCES manufacturers(id));
```

Q #9) What are the differences between CHAR and VARCHAR data types?

Answer:

Both CHAR and VARCHAR data types are used to store string data in the field of the table.

The differences between these data types are mentioned below:

- CHAR data type is used to store fixed-length string data and VARCHAR data type is used to store variable-length string data.
- The storage size of CHAR data type will always be the maximum length of this data type and the storage size of VARCHAR will be the length of the inserted string data. Hence, it is better to use the CHAR data type when the length of the string will be the same length for all the records.
- CHAR is used to store small data whereas VARCHAR is used to store large data.
- CHAR works faster and VARCHAR works slower.

Example:

The following SQL statement will create a table named customers. In this table, the data type of **name** field is VARCHAR and the data type of **phone** field is CHAR.

The size of the **name** field will depend on the length of the inserted value. The size of the **phone** field will always be 14 characters even if the length of the inserted value is less than 14 characters.

```
1  CREATE TABLE customers (
2      id INT AUTO_INCREMENT PRIMARY KEY,
3      name VARCHAR(50),
4      phone CHAR(14))
```

Q #10) What is the purpose of using TIMESTAMP data type?

Answer:

A TIMESTAMP data type is used to store the combination of date and time value which is 19 characters long.

The format of TIMESTAMP is YYYY-MM-DD HH:MM: SS. It can store data from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC. By default, the current date and time of the server get inserted in the field of this data type when a new record is inserted or updated.

Q #11) What is the difference between mysql_fetch_array() and ysql_fetch_object() ?

Answer:

Both mysql_fetch_array() and mysql_fetch_object() are built-in methods of PHP to retrieve records from MySQL database table.

The difference between these methods is that mysql_fetch_array() returns the result set as an array and mysql_fetch_object() returns the result set as an object.

Example:

```
 1  $result = mysql_query("SELECT id, name FROM clients");
 2
 3  //using mysql_fetch_array()
 4  while ($row = mysql_fetch_array($result, MYSQL_NUM)) {
 5      printf("ID: %s  Name: %s", $row[0], $row[1]);
 6  }
 7
 8  //using mysql_fetch_object()
 9  while ($row = mysql_fetch_object($result)) {
10      printf("ID: %s  Name: %s", $row->id, $row->name);
11  }
```

Q #12) How can you filter the duplicate data while retrieving records from the table?

Answer:

A DISTINCT keyword is used to filter the duplicate data from the table while retrieving the records from a table.

Example:

The following SQL command shows all the records of the **items** table. The output shows that the table contains duplicate values in the **type** field.

SELECT * from items;

The following SQL command will display the values of **type** field by removing the duplicate values.

SELECT DISTINCT type from items;

Q #13) What is the difference between NOW() and CURRENT_DATE()?

Answer:

Both **NOW()** and **CURRENT_DATE()** are built-in MySQL methods. **NOW()** is used to show the current date and time of the server and **CURRENT_DATE()** is used to show only the date of the server.

SELECT now();

SELECT current_date();

Q #14) Which statement is used in a select query for partial matching?

Answer:

REGEXP and **LIKE** statements can be used in a select query for partial matching. REGEXP is used to search records based on the pattern and LIKE is used to search any record by matching any string at the beginning or end or middle of a particular field value.

Example:

First, check the existing records of the '**clients**' table by executing the select query.

SELECT * FROM clients;

Run SELECT query with REGEXP clause to search those records from the **clients** where the client name starts with '**S**'

SELECT * FROM clients WHERE name REGEXP "^S";

Run SELECT query with LIKE clause to search those records from the **clients** where the client name starts with '**A**'

SELECT * FROM clients WHERE name LIKE "A%";

Q #15) Which MySQL function is used to concatenate string?

Answer:

CONCAT() function is used to combine two or more string data. The use of this function is here with an example.

Example:

The following SELECT query with CONCAT() function will combine five words, 'Welcome ', 'to', 'SoftwareTestingHelp','.' and 'com'.

SELECT CONCAT('Welcome ','to ','SoftwareTestingHelp','.','com');

CONCAT() function can be used on any table as well. The following SELECT query will show the output by combining two fields, **brand** and **type** of **items** table.

SELECT CONCAT(brand,'=>',type) from items;

Q #16) How can you change the name of any existing table by using the SQL statement?

Answer:

The following SQL command is used to rename an existing table of the database.

RENAME TABLE table_name TO new_name

<u>**Example:**</u>

The following command will show the table list of the **newdb** database.

SHOW TABLES;

The following rename command will rename the table **items** by new name **products**.

RENAME TABLE items TO products;

SHOW TABLES;

Q #17) How can you retrieve a portion of any column value by using a select query?

Answer:

SUBSTR() function is used to retrieve the portion of any column. The use of this function is explained here with an example.

Example:

Here, the first select command is used to show all the records of the products table and the second select command is executed using SUBSTR function and that prints only the first five characters of the name field.

SELECT * FROM products;

SELECT SUBSTR(name,1,5) FROM products;

Q #18) What is the purpose of using a HEAP table?

Answer:

The table which uses a hashed index and stores in the memory is called HEAP table. It works as a temporary table and it uses the indexes that make it faster than another table type.

When MySQL crashes for any reason then all the data stored in this table can be lost. It uses fixed length data types. Hence BLOB and TEXT data types are not supported by this table. It is a useful table for those MySQL tasks where speed is the most important factor and temporary data is used.

Q #19) How can you add and remove any column of a table?

Answer:

The syntax for adding any column in an existing table is shown below.

ALTER TABLE table_name ADD COLUMN column_name column_definition [FIRST|AFTER existing_column]

<u>**Example:**</u>

DESCRIBE command is used to show the structure of the products table.

DESCRIBE products;

The following ALTER command with ADD COLUMN clause will add a new field named '**price**' in the table **products**.

ALTER TABLE products ADD COLUMN price DECIMAL(5,2);

DESCRIBE products;

The syntax for removing any column from an existing table is shown below.

ALTER TABLE table_name DROP COLUMN column_name;

<u>**Example:**</u>

The following ALTER command with a DROP COLUMN clause will remove the field named '**brand'** in the table '**products'**.

ALTER TABLE products DROP COLUMN brand;

DESCRIBE products;

Q #20) What is an index? How can an index be declared in MySQL?

Answer:

An index is a data structure of MySQL table that is used to speed up the queries.

It is used by the database search engine to find out the records faster. One or more fields of a table can be used as an index key. Index key can be assigned at the time of table declaration or can be assigned after creating the table.

<u>**Example:**</u>

username and **email** fields are set as the index in the following create table statement.

```
1  CREATE TABLE users(
2      username VARCHAR(50) PRIMARY KEY,
3      email VARCHAR(50) NOT NULL,
4      password VARCHAR(50) NOT NULL,
5      INDEX (username, email));
```

The following command will show the index key information of the '**users'** table.

SHOW INDEXES FROM users;

Q #21) What is meant by decimal (5,2)?

Answer:

A decimal data type is used in MySQL to store the fractional data.

The meaning of decimal (5,2) means that the total length of the fractional value is 5. The field can contain 3 digits before the decimal point and 2 digits after the decimal point. If a user adds any value larger than the defined length then it will insert 999.99 in the field.

The use of this data type is explained in the following example.

Example:

In the following insert query, **789.56** is inserted in the **price** field. This value is less than 1000 and the total digits with the fractional part are 5. So, this value is valid for this field.

```
1  INSERT INTO products (type, name, price, manufacturer_id)
2  VALUES ('Mobile', 'iPhone 8', 789.56, 1);
3  SELECT * FROM products;
```

In the following insert query, **34789.567** is set for the price field. Then this value is greater than 1000 and the total digits with fractional part are 8. So, the default value 999.99 is inserted in the place of 34789.567**.**

```
1  INSERT INTO products (type, name, price, manufacturer_id)
2  VALUES('TV','Sony 32" Smart TV',34789.567, 2);
3  SELECT * FROM products;
```

Q #22) What is view? How can you create and drop view in MySQL?

Answer:

A view works as a virtual table that is used to store query and returns a result set when it is called. An updatable view is also supported by MySQL.

The ways in which a view can be created or deleted in MySQL is shown in the following examples.

Create View Example:

The following statement will create a view file named '**client_list**' based on the table **clients**.

CREATE VIEW `client_list` AS SELECT `name` as 'Name', `membership` as 'Membership' FROM `clients`;

Select statement will display the records of **client_list** value.

SELECT * FROM client_list;

Drop View Example:

Drop view statement will delete the view file. Select query will show an error after deleting the view.

DROP VIEW client_list;

SELECT * FROM client_list;

Q #23) What is the function of mysqldump?

Answer:

mysqldump is an useful utility tool of MySQL that is used to dump one or more or all databases from the server for backup or transfer to another database server.

Syntax:

For a single database,

mysqldump [OPTIONS] db_name [TABLES]

For multiple databases,

mysqldump [OPTIONS] –databases DB1 [DB2 DB3...]

For all databases,

mysqldump [OPTIONS] –all-databases

Example:

The following command will create a dump of the **'newdb'** database and export the content of the database in the file, **newdb.sql**.

mysqldump –databases newdb > newdb.sql

Q #24) How can you change the password of a MySQL user?

Answer:

SET **PASSWORD** statement is used to change the password of a MySQL user.

Syntax:

SET PASSWORD FOR 'username'@'hostname' = PASSWORD('password');

Example:

The following statement will set or change the root password.

SET PASSWORD FOR 'root'@'localhost' = PASSWORD('123456');

Q #25) What is the difference between UNIX TIMESTAMP and MySQL TIMESTAMP?

Answer:

Both UNIX TIMESTAMP and MySQL TIMESTAMP are used to represent the date and time value. The main difference between these values is that UNIX TIMESTAMP represents the value by using 32-bits integers and MySQL TIMESTAMP represents the value in the human-readable format.

Example:

A UNIX time value is used by FROM_UNIXTIME function in SELECT query to get the date and time value in the human-readable format.

SELECT FROM_UNIXTIME (1596222320) AS 'MySQLTIMESTAMP';

Date and time value is used by UNIX_TIMESTAMP function in SELECT query to get the date and time value in the UNIX format.

SELECT UNIX_TIMESTAMP ('2018-12-25 09:45:40') AS 'UNIXTIMESTAMP';

Q #26) How can you import tables from a SQL file into a database by using the MySQL client?

Answer:

Database tables can be imported into a database from a SQL file by using the following MySQL statement.

mysql -u username -p database_name < sql_filename

<u>**Example:**</u>

If the root user's password is empty, then the following command will import tables from 'newdb.sql' file into the database **`mydb`.**

mysql -u root mydb < newdb.sql

Q #27) What is the difference between Primary key and Unique key?

Answer:

Unique data is stored in the primary key and unique key fields. Primary key field never accepts NULL value but unique key field accepts a NULL value.

Example:

In the **users'** table, **id** field is the **primary key** and **email** field is a **unique key**. Two records are inserted in the table where the email field is NULL for the 2nd record. The records are inserted properly as the unique field supports a NULL value.

```
1  INSERT INTO users (username, email, password)
2  VALUES('admin', 'admin@example.com', '7890'),
3              ('staff', 'NULL', '1234');
4  SELECT * FROM users;
```

Q #28) What is the purpose of using IFNULL() function?

Answer:

IFNULL() function takes two arguments. It returns the first argument value if the value of the first argument is not NULL and it returns the second argument if the value of the first argument is NULL.

Example:

Here, the first argument of IFNULL function is not NULL. So, the output is the first argument value.

SELECT IFNULL ("Tutorial", "fahmidasclassroom.com");

Here, the first argument of IFNULL function is NULL. So, the output is NULL.

SELECT IFNULL ("NULL", "fahmidasclassroom.com");

Q #29) What is a join? Explain the different types of MySQL joins.

Answer:

The SQL statement that is used to make a connection between two or more tables based on the matching columns is called a join. It is mainly used for complex queries.

Different types of SQL joins are mentioned below:

- **Inner Join**: It is a default join. It returns records when the values match in the joining tables.
- **Left Outer Join**: It returns all the records from the left table based on the matched records from the right table.
- **Right Outer Join**: It returns all the records from the right table based on the matched records from the left table.
- **Full Outer Join**: It returns all the records that match from the left or right table.

<u>Example:</u>

Two tables, **manufacturers** and **products** are used in this example to show the use of INNER JOIN. Here, SELECT queries are used to show the current records of these two tables.

SELECT * FROM manufacturers;

SELECT * FROM products;

INNER JOIN is used in the following SELECT query where all the id and name of products table will be displayed based on matching **manufacturer_id** of the **products** with an **id** of the **manufacturers** table.

```
1  SELECT products.id, products.name
2  FROM products
3  INNER JOIN manufacturers ON manufacturers.id= products.manufacturer_id;
```

Q #30) How can you retrieve a particular number of records from a table?

Answer:

LIMIT clause is used with the SQL statement to retrieve a particular number of records from a table. From which record and how many records will be retrieved are defined by the LIMIT clause.

Syntax:

LIMIT starting_number, number_of_rows

Example:

Products table has 5 records which are displayed by the first select query and the second select query is used to display the records from 2nd to 3rd by using LIMIT 1, 2.

SELECT * FROM products;

SELECT * FROM products LIMIT 1, 2;

Answer:

'-X' option is used with `mysql` command for exporting the file as XML. The following statement will export any table from a database as an XML file.

mysql -u username -X -e "SELECT query" database_name

<u>**Example:**</u>

The following command will export the data of the **items** table into **an xmlData.xml** file.

mysql -u root -X -e "SELECT * from products" newdb > xmlData.xml

Q #32) What is a CSV table?

Answer:

MySQL table that uses CSV storage engine is called a CSV table. Data are stored as comma-separated values in the CSV table. MySQL server creates a data file with an extension '.csv' to store the content of the CSV table.

Example:

The following create statement will create a CSV file named book.

CREATE TABLE book (id INT NOT NULL) ENGINE=CSV;

Q #33) How can you calculate the sum of any column of a table?

Answer:

SUM() function is used to calculate the sum of any column.

Syntax:

SUM(DISTINCT expression)

Example:

Products table has a numeric field named, price. In this example, the **SUM() function** is used to calculate the total value of the **price** field.

```
1   SELECT * FROM products;
2   SELECT SUM(price) as total FROM products;
```

Q #34) How can you count the total number of records of any table?

Answer:

COUNT() function is used to count the total number of records of any table.

Syntax:

COUNT(expression)

Example:

The following select query is used to count the total number of records of the **products** table.

SELECT COUNT(*) as `Total Records` FROM products;

Q #35) Explain the difference between delete and truncate.

Answer:

Both DELETE and TRUNCATE commands are used to delete the records from any database table. However, there are some significant differences between these commands. If the table contains AUTO_INCREMENT PRIMARY KEY field then the effect of these commands can be shown properly.

Two differences between these commands are mentioned below.

#1) DELETE command is used to delete a single or multiple or all the records from the table. TRUNCATE command is used to delete all the records from the table or make the table empty.

#2) When DELETE command is used to delete all the records from the table then it doesn't re-initialize the table. So, AUTO_INCREMENT field does not count from one when the user inserts any record.

But when all the records of any table are deleted by using TRUNCATE command then it re-initializes the table and a new record will start from one for the AUTO_INCREMENT field.

Example:

Previously created users table is used in this example.

First, the SELECT query will show all the records of the users table. DELETE query will delete all the records from the user's table. INSERT query will insert a new record into the users table. After insert, if the SELECT query executes again then it will be shown that a new **id** is calculated after the deleted **id**.

```
1  SELECT * FROM users;
2  DELETE FROM users;
3  INSERT INTO users (username, email)
4  VALUES ('Durjoy', 'durjoy@gmail.com');
5  SELECT * FROM users;
```

Currently, there are two records in the users table and when a new record is inserted after deleting all the records then the new id is 3, and not 1.

Same queries are executed in this part, just used TRUNCATE statement in place of DELETE. It is shown that the id value of the new record is 1.

```
1  TRUNCATE table users;
2  INSERT INTO users (username, email)
3  VALUES ('Farheen', 'farheen@gmail.com');
4  SELECT * FROM users;
```

Q #36) What is a storage engine? What are the differences between InnoDB and MyISAM engines?

Answer:

One of the major components of the MySQL server is the storage engine for doing different types of database operations. Each database table created is based on the specific storage engine.

MySQL supports two types of storage engines i.e **transactional and non-transactional**. InnoDB is the default storage engine of MySQL which is a transactional storage engine. MyISAM storage engine is a non-transactional storage engine.

The differences between InnoDB and MyISAM storage engines are discussed below:

- MyISAM supports FULLTEXT index but InnoDB doesn't support FULLTEXT index.
- MyISAM is faster and InnoDB is slower.
- InnoDB supports ACID (Atomicity, Consistency, Isolation, and Durability) property but MyISAM doesn't.
- InnoDB supports row-level locking and MyISAM support table-level locking.
- InnoDB is suitable for large database and MyISAM is suitable for a small database.

Q #37) What is a transaction? Describe MySQL transaction properties.

Answer:

When a group of database operations is done as a single unit then it is called a transaction. If any task of the transactional tasks remains incomplete then the transaction will not succeed. Hence, it

is mandatory to complete all the tasks of a transaction to make the transaction successful.

A transaction has four properties which are known as ACID property. These properties are described below.

- **Atomicity:** It ensures that all the tasks of a transaction will be completed successfully otherwise all the completed tasks will be rolled back to the previous state for any failure.
- **Consistency:** It ensures that the database state must be changed accurately for the committed transaction.
- **Isolation:** It ensures that all the tasks of a transaction will be done independently and transparently.
- **Durability:** It ensures that all the committed transaction is consistent for any type of system failure.

Q #38) What are the functions of commit and rollback statements?

Answer:

Commit is a transaction command that executes when all the tasks of a transaction are completed successfully. It will modify the database permanently to confirm the transaction.

Syntax:

COMMIT;

Rollback is another transactional command that executes when any of the transactional tasks becomes unsuccessful and undoes all the changes that are made by any transactional task to make the transaction unsuccessful.

Syntax:

ROLLBACK;

Q #39) What is the difference between MyISAM static and MyISAM dynamic?

Answer:

MyISAM static and MyISAM dynamic are the variations of the MyISAM storage engine. The differences between these tables are mentioned below.

- All the fields of MyISAM static table are of a fixed length and MyISAM dynamic table accepts variable length fields such as BLOB, TEXT etc.
- After data corruption, it is easier to restore MyISAM static table than MyISAM dynamic table.

Q #40) What is a trigger? How you can create a trigger in MySQL?

Answer:

One of the important features of the MySQL database is a trigger that executes automatically when a particular database event occurs.

It fires after or before the execution of an insert or update or delete statement. It is a very useful option when a database user wants to do some database operations automatically.

Trigger Example:

If you want to delete the items of a supplier from the **items** table automatically after deleting the entry of the particular supplier from the '**suppliers**' table then write the trigger in the following way.

Example:

This is an example of after delete trigger that will fire automatically when any record is removed from the **manufacturer** table and deletes all the records from the **products** table where the deleted **id** of the **manufacturer** table matches with the **manufacturer_id** field of the **products** table.
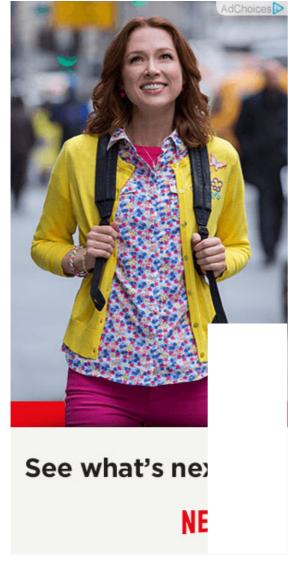
```
1 DELIMITER //
2 CREATE TRIGGER manufacturer_after_delete
3 AFTER DELETE
4 ON manufacturers FOR EACH ROW
5 BEGIN
6 DELETE FROM products WHERE products.manufacturers_id = OLD.id;
7 END;
8 //
```

Conclusion

MySQL server has several built-in functions and clauses to perform different types of actions on the table data. Most commonly used SQL functions and clauses of the MySQL server are explained in this article with different examples.

Hope, this article on the best MySQL interview questions will help you to understand the basic as well as advanced level concepts of the MySQL server for fresher as well as the experienced professionals.

We wish you all the best!!

Join Over 200,000+ Testers

Get premium ebooks and testing tips.

Enter your email here...

SUBSCRIBE NOW!

Join Over 300000+ Followers!

Top FREE Training Tutorials

Load Testing Buyer's Guide

Free QA Training

Selenium Tutorials

QTP/UFT Tutorials

ISTQB Free Study Guide

ISTQB Premium Study Guide

Quality Center Tutorials

LoadRunner Tutorials

JMeter Tutorials

JIRA Tutorials

VBScript Tutorials

Best Test Management Tools
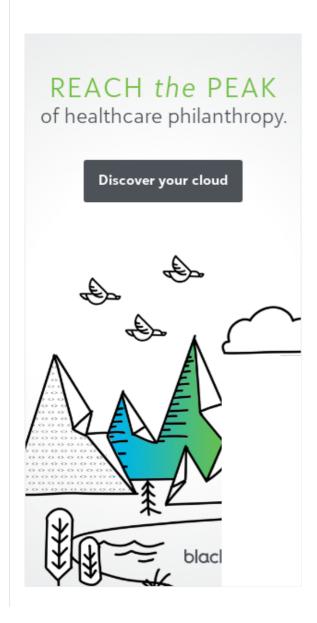
Appium Studio Tutorials

Unix Tutorials

Free DevOps Training

Free JAVA Tutorials

Free Python Tutorials

101+ Interview Questions

Adv

Thanks For Reading! 50 Million Visits and Counting!