

SQL - Using Sequences

A sequence is a set of integers 1, 2, 3, ... that are generated in order on demand. Sequences are frequently used in databases because many applications require each row in a table to contain a unique value and sequences provide an easy way to generate them.

This chapter describes how to use sequences in MySQL.

Using AUTO_INCREMENT column

The simplest way in MySQL to use sequences is to define a column as AUTO_INCREMENT and leave the rest to MySQL to take care.

Example

Try out the following example. This will create a table and after that it will insert a few rows in this table where it is not required to give a record ID because its auto-incremented by MySQL.

```
mysql> CREATE TABLE INSECT
-> (
-> id INT UNSIGNED NOT NULL AUTO_INCREMENT,
-> PRIMARY KEY (id),
-> name VARCHAR(30) NOT NULL, # type of insect
-> date DATE NOT NULL, # date collected
-> origin VARCHAR(30) NOT NULL # where collected
);
Query OK, 0 rows affected (0.02 sec)
mysql> INSERT INTO INSECT (id,name,date,origin) VALUES
-> (NULL,'housefly','2001-09-10','kitchen'),
-> (NULL,'millipede','2001-09-10','driveway'),
-> (NULL,'grasshopper','2001-09-10','front yard');
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> SELECT * FROM INSECT ORDER BY id;
```

id	name	date	origin
1	housefly	2001-09-10	kitchen
2	millipede	2001-09-10	driveway
3	grasshopper	2001-09-10	front yard

```
3 rows in set (0.00 sec)
```

Obtain AUTO_INCREMENT Values

The `LAST_INSERT_ID()` is an SQL function, so you can use it from within any client that understands how to issue SQL statements. Otherwise PERL and PHP scripts provide exclusive functions to retrieve auto-incremented value of last record.

PERL Example

Use the **`mysql_insertid`** attribute to obtain the `AUTO_INCREMENT` value generated by a query. This attribute is accessed through either a database handle or a statement handle, depending on how you issue the query. The following example references it through the database handle.

```
$dbh->do ("INSERT INTO INSECT (name,date,origin)
VALUES('moth','2001-09-14','windowsill')");
my $seq = $dbh->{mysql_insertid};
```

PHP Example

After issuing a query that generates an `AUTO_INCREMENT` value, retrieve the value by calling the **`mysql_insert_id()`** function.

```
mysql_query ("INSERT INTO INSECT (name,date,origin)
VALUES('moth','2001-09-14','windowsill')", $conn_id);
$seq = mysql_insert_id ($conn_id);
```

Renumbering an Existing Sequence

There may be a case when you have deleted many records from a table and you want to re-sequence all the records. This can be done by using a simple trick, but you should be very careful to do this and check if your table is having a join with another table or not.

If you determine that resequencing an `AUTO_INCREMENT` column is unavoidable, the way to do it is to drop the column from the table, then add it again.

The following example shows how to renumber the id values in the insect table using this technique.

```
mysql> ALTER TABLE INSECT DROP id;
mysql> ALTER TABLE insect
...-> ADD id INT UNSIGNED NOT NULL AUTO_INCREMENT FIRST,
...-> ADD PRIMARY KEY (id);
```

Starting a Sequence at a Particular Value

By default, MySQL will start the sequence from 1, but you can specify any other number as well at the time of table creation.

The following code block has an example where MySQL will start sequence from 100.

```
mysql> CREATE TABLE INSECT
... -> (
... -> id INT UNSIGNED NOT NULL AUTO_INCREMENT = 100,
... -> PRIMARY KEY (id),
... -> name VARCHAR(30) NOT NULL, # type of insect
... -> date DATE NOT NULL, # date collected
... -> origin VARCHAR(30) NOT NULL # where collected
... );
```

Alternatively, you can create the table and then set the initial sequence value with ALTER TABLE.

```
mysql> ALTER TABLE t AUTO_INCREMENT = 100;
```