Table of Contents

DBMS vary in approaches to generating values for inclusion in tables. MySQL has an option you can set for a column in a table that generates new integer values for each new row inserted into the table. We will first look at this in its simplest format for simply generating numbers; then we will look at some of the complications.

I have not used autoincrement columns in any of the tables we use because I want more control over the id values, but many people use autoincrement column so you need to have seen this.

Suppose we have a table that is a stand-alone table- it does not have any relationships to other tables. We want to have an ID column and we want to use a surrogate key. A surrogate key is one that does not have any dependence on naturally occurring data- it is not the customer's phone number, it is not the employee's SSN or the book's ISBN. It is just a number. Most commonly this would be a set of values 1, 2, 3, 4, etc. This is where the autoincrement feature works the best.

1. Creating an Autoincrement

If you work through this set of demos be careful to do each of them in order- with an autoincrement you can run some additional statements and end up with a different sequence of numbers.

Demo 01: Create table with an auto_increment and Insert rows.

```
Create table z_autoIncr_1 (
   col_1 integer unsigned auto_increment primary key
,  col_2 varchar(15)) engine= innodb;

insert into z_autoIncr_1 (col_2 ) values ('Ann');
insert into z_autoIncr_1 (col_2 ) values ('Betsy');
insert into z_autoIncr_1 (col_2 ) values ('Carla');
insert into z_autoIncr_1 (col_2 ) values ('Darlene');

select * from z_autoIncr_1;
+-------+---------+
| col_1 | col_2   |
+-------+---------+
|     1 | Ann     |
|     2 | Betsy   |
|     3 | Carla   |
|     4 | Darlene |
+-------+---------+
```

For each new row, the value of col_1 is incremented by one.

You can have only one auto_increment attribute per table.

Demo 02: You can also use a null for the auto_increment attribute and get the generated value.

```
insert into z_autoIncr_1 values (null, 'Eartha');
insert into z_autoIncr_1 values (null, 'Fanny');
+-------+---------+
| col_1 | col_2   |
+-------+---------+
|     1 | Ann     |
|     2 | Betsy   |
|     3 | Carla   |
|     4 | Darlene |
|     5 | Eartha  |
|     6 | Fanny   |
+-------+---------+
```

Demo 03: Suppose we delete a row.

```
delete from z_autoIncr_1 where col_1 = 4;
+-------+---------+
| col_1 | col_2   |
+-------+---------+
|     1 | Ann     |
|     2 | Betsy   |
|     3 | Carla   |
|     5 | Eartha  |
|     6 | Fanny   |
+-------+---------+
```

Demo 04: And then add back the row for Darlene. This does not fill in the gaps. The new row gets the next number.

```
insert into z_autoIncr_1 (col_2 ) values ('Darlene');
+-------+---------+
| col_1 | col_2   |
+-------+---------+
|     1 | Ann     |
|     2 | Betsy   |
|     3 | Carla   |
|     5 | Eartha  |
|     6 | Fanny   |
|     7 | Darlene |
+-------+---------+
```

If we add a row with a specific value for col_1 and then go back to the auto_increment, we will get a gap in the numbers.

Demo 05: Add a row with col_1 set to 66. And then auto_increment rows and they get the next numbers in the sequence.

```
insert into z_autoIncr_1 values (66, 'Annabelle');
insert into z_autoIncr_1 values (null, 'Barbar');
insert into z_autoIncr_1 values (null, 'Curtis');

select * from z_autoIncr_1;
+-------+-----------+
| col_1 | col_2     |
+-------+-----------+
|     1 | Ann       |
|     2 | Betsy     |
|     3 | Carla     |
|     5 | Eartha    |
|     6 | Fanny     |
|     7 | Darlene   |
|    66 | Annabelle |
|    67 | Barbar    |
|    68 | Curtis    |
+-------+-----------+
```

The use of the autoincrement attribute is the easiest to understand if you use it to simply create a series of ID that will be numbered 1, 2, 3, etc.

The rest of this is optional but if you are going to use an autoincrement column then you do need to consider these issues.

2. Setting an increment and an offset

You have the option of setting an increment, and with some restrictions, an offset.

Demo 06:

```
Set @@auto_increment_increment = 5;

Create table z_autoIncr_2 (
   col_1 integer unsigned auto_increment primary key
,  col_2 varchar(15)) engine= innodb;

insert into z_autoIncr_2 (col_2 ) values ('ant');
insert into z_autoIncr_2 (col_2 ) values ('beetle');
insert into z_autoIncr_2 (col_2 ) values ('cricket');
insert into z_autoIncr_2 (col_2 ) values ('dragonfly');
select * from z_autoIncr_2;
+-------+-----------+
| col_1 | col_2     |
+-------+-----------+
|     1 | ant       |
|     6 | beetle    |
|    11 | cricket   |
|    16 | dragonfly |
+-------+-----------+
```

Demo 07: But that setting applies to all tables we have with an autoincrement - so this is rather dangerous to use if you are not careful. Insert a few more rows to the first table.

```
insert into z_autoIncr_1 values (null, 'Douglas');  -- this gets +3
insert into z_autoIncr_1 values (null, 'Evens');    -- this gets +5
insert into z_autoIncr_1 values (null, 'Frank');    -- this gets +5
insert into z_autoIncr_1 values (null, 'George');   -- this gets +5

select * from z_autoIncr_1;
+--------+-----------+
| col_1  | col_2     |
+--------+-----------+
|      1 | Ann       |
|      2 | Betsy     |
|      3 | Carla     |
|      5 | Eartha    |
|      6 | Fanny     |
|      7 | Darlene   |
|     66 | Annabelle |
|     67 | Barbar    |
|     68 | Curtis    |
|     71 | Douglas   |
|     76 | Evens     |
|     81 | Frank     |
|     86 | George    |
+-------+-----------+
```

Demo 08: Let's try another pair of settings

```
Set @@auto_increment_offset =25, @@auto_increment_increment = 25;

Create table z_autoIncr_3 (
   col_1 integer unsigned auto_increment primary key
,  col_2 varchar(15)) engine= innodb;

insert into z_autoIncr_3 (col_2 ) values ('Ann');
insert into z_autoIncr_3 (col_2 ) values ('Betsy');
insert into z_autoIncr_3 (col_2 ) values ('Carla');
insert into z_autoIncr_3 (col_2 ) values ('Darlene');

select * from z_autoIncr_3;
+-------+---------+
| col_1 | col_2   |
+-------+---------+
|    25 | Ann     |
|    50 | Betsy   |
|    75 | Carla   |
|   100 | Darlene |
+-------+---------+
```

For this to work properly the value of the offset must be <= increment. This is not obvious! And not very practical.

Demo 09: One more example of autoincrement oddities. Try running this and see if you can see what is happening. This is version specific so you might not even get the same results.

```
set @@auto_increment_offset = 1, @@auto_increment_increment = 1;

create table z_autoincr_4 (
 col_1 int not null auto_increment primary key,
 col_2 int not null
) engine= innodb;

insert into z_autoincr_4 (col_1, col_2) values (0, 101);
insert into z_autoincr_4 (col_1, col_2) values (-1, 102);
insert into z_autoincr_4 (col_1, col_2) values (9999999999, 103);
```

```
Error: Out of range value adjusted for column 'Col_1' at row 1
```

```
insert into z_autoincr_4  col_1, col_2) values (1, 104);
```

```
Error:Duplicate entry '1' for key 1
```

```
insert into z_autoincr_4 (col_1, col_2) values (0, 105);
insert into z_autoincr_4 (col_1, col_2) values (2147483647, 106);

select * from z_autoincr_4;
+------------+-------+
| col_1      | col_2 |
+------------+-------+
|         -1 |   102 |
|          1 |   101 |
|          2 |   105 |
| 2147483647 |   106 |
+------------+-------+
```

Trying to insert a value of 0 for the autoincrement column gives you the next number in the sequence- even though a 0 is not a null.

Demo 10: Resetting the autoincrement value

Let's make one more table; insert a few rows and then run an Alter table query and insert more rows. This lets you reset the starting value. You can use this technique with an empty table to get the starting number your want.

```
create table z_autoincr_5 (
 col_1 int not null auto_increment primary key,
 col_2 int not null
) engine= innodb;


insert into z_autoincr_5  (col_1, col_2) values (0, 101);
insert into z_autoincr_5  (col_1, col_2) values (null, 102);
insert into z_autoincr_5  (col_1, col_2) values (0, 103);
insert into z_autoincr_5  (col_1, col_2) values (0, 104);


alter table z_autoincr_5 auto_increment = 500;
insert into z_autoincr_5  (col_1, col_2) values (0, 105);
insert into z_autoincr_5  (col_1, col_2) values (0, 106);

select * from z_autoincr_5;
+-------+-------+
| col_1 | col_2 |
+-------+-------+
|     1 |   101 |
|     2 |   102 |
|     3 |   103 |
|     4 |   104 |
|   500 |   105 |
|   501 |   106 |
+-------+-------+
```