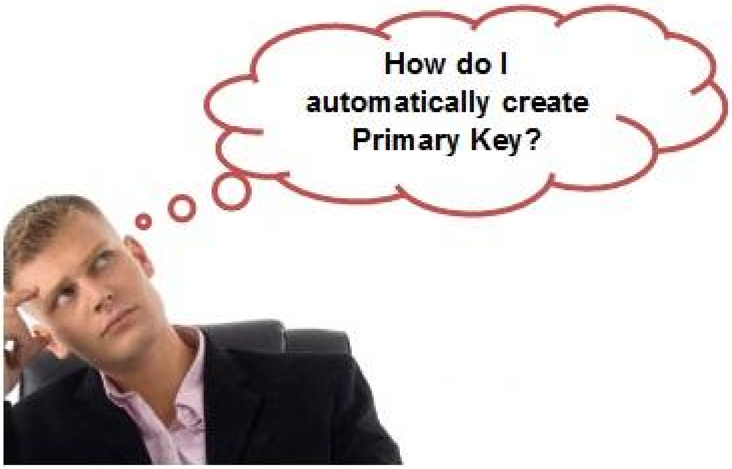
MySQL AUTO_INCREMENT with Examples

What is auto increment?

Auto Increment is a function that operates on numeric data types. It automatically generates sequential numeric values every time that a record is inserted into a table for a field defined as auto increment.

When to use auto increment?

In the lesson on database normalization, we looked at how data can be stored with minimal redundancy, by storing data into many small tables ,related to each other using primary and foreign keys.



A primary key must be unique as it uniquely identifies a row in a database. But, how can we ensure that the primary key is always unique? One of the possible solutions would be, to use a formula to generate the primary key, which checks for the existence of the key, in the table, before adding data. This may work well but as you can see the approach is complex and not foolproof. In order to avoid such complexity and to ensure that the primary key is always unique, we can use MySQL's Auto increment feature to generate primary keys. Auto increment is used with the INT data type. The INT data type supports both signed and unsigned values. Unsigned data types can only contain positive numbers. As a best practice, it is recommended to define the unsigned constraint on the auto increment primary key.

Auto increment syntax

Let's now look at the script used to create the movie categories table.

```
CREATE TABLE `categories` (
  `category_id` int(11) AUTO_INCREMENT,
  `category_name` varchar(150) DEFAULT NULL,
  `remarks` varchar(500) DEFAULT NULL,
  PRIMARY KEY (`category_id`)
);
```

Notice the "AUTO_INCREMENT" on the category_id field. This causes the category Id to be automatically generated every time a new row is inserted into the table. It is not supplied when inserting data into the table, MySQL generates it.

By default, the starting value for AUTO_INCREMENT is 1, and it will increment by 1 for each new record

Let's examine the current contents of the categories table.

```
SELECT * FROM `categories`;
```

Executing the above script in MySQL workbench against the myflixdb gives us the following results.

category_id	**category_name**	**remarks**
1	Comedy	Movies with humour
2	Romantic	Love stories

3	Epic	Story ancient movies
4	Horror	NULL
5	Science Fiction	NULL
6	Thriller	NULL
7	Action	NULL
8	Romantic Comedy	NULL

Let's now insert a new category into the categories table .

```
INSERT INTO  `categories` (`category_name`) VALUES
('Cartoons');
```

Executing the above script against the myflixdb in MySQL workbench gives us the following results shown below.

category_id	**category_name**	**remarks**
1	Comedy	Movies with humour
2	Romantic	Love stories
3	Epic	Story ancient movies
4	Horror	NULL
5	Science Fiction	NULL
6	Thriller	NULL
7	Action	NULL

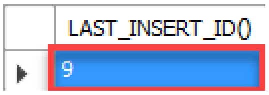
8	Romantic Comedy	NULL
9	Cartoons	NULL

Note we didn't supply the category id. MySQL automatically generated it for us because the category id is defined as auto increment.

If you want to get the last insert id that was generated by MySQL, you can use the LAST_INSERT_ID function to do that. The script shown below gets the last id that was generated.

```
SELECT LAST_INSERT_ID();
```

Executing the above script gives the last Auto increment number generated by the INSERT query. The results are shown below.



LAST_INSERT_ID()
9

Summary

- Auto increment attribute when specified on a column with a numeric data type, generates numbers sequentially whenever a new row is added into the database.
- The Auto increment is commonly used to generate primary keys.
- The defined data type on the Auto increment should be large enough to accommodate many records. Defining TINYINT as the data type for an auto increment field limits the number of records that can be added to the table to 255 only since any values beyond that would not be accepted by the TINYINT data type.
- It is considered a good practice to specify the unsigned constraint on auto increment primary keys to avoid having negative numbers.
- When a row is deleted from a table, its auto incremented id is not re-used. MySQL continues generating new numbers sequentially.
- By default, the starting value for AUTO_INCREMENT is 1, and it will increment by 1 for each new record

- To let AUTO_INCREMENT sequence start with another value , use AUTO_INCREMENT = 10