

AutoIncrement Primary Key

<http://sqldatabase.net/docs/autoincrement.aspx>

When auto increment property is set on column it will generate sequential unique numeric values automatically whenever a new row is inserted in that table. It works only with numeric data types and in most cases it must be used with primary key constraints. The value starts at one (1) and incremented with 1. The last generated value is stored in sys_sequences table.











```
DROP TABLE IF EXISTS AutoIncrementTest;CREATE TABLE AutoIncrementTest(Id Integer Primary Key AUTOINCREMENT,Name Text);
```

Values can be added by specifying if you want library to generate the value then you can either omit the column or pass null value.

```
Insert Into AutoIncrementTest (Name) Values ('Mary'); /* column supplied */
Insert Into AutoIncrementTest Values (3, 'John'); /* Value supplied */
Insert Into AutoIncrementTest Values (5, 'David'); /* Value supplied */
Insert Into AutoIncrementTest Values (null, 'Smith'); /* No Value */
Insert Into AutoIncrementTest Values (null, 'Robert'); /* No Value */

SELECT * FROM AutoIncrementTest;
```

Colors are marked in the image below for clarity the red , blue and black representing three different type of insert statements above.

	Id	Name
▶ 	1 	Mary
	3 	John
	5 	David
	6 	Smith
	7 	Robert

Update or Set Value

You can also set start value after sequence has been generated by updating the sys_sequences table.

```
UPDATE sys_sequences SET lastnumber = '10' WHERE TableName = 'AutoIncrementTest';
Insert Into AutoIncrementTest Values (null, 'Scott'); SELECT * FROM AutoIncrementTest;
```

	Id	Name
►	1	Mary
	3	John
	5	David
	6	Smith
	7	Robert
	11	Scott

Delete or Reset AutoIncrement.

If you delete the row in sys_sequences table then the maximum value will be taken to restart the sequence, it is useful if you have deleted all the rows and would like to reset the seed.

```
DELETE FROM sys_sequences WHERE TableName = 'AutoIncrementTest';
Insert Into AutoIncrementTest Values (null, 'Peter');

SELECT * FROM AutoIncrementTest;
```

	Id	Name
►	1	Mary
	3	John
	5	David
	6	Smith
	7	Robert
	11	Scott
	12	Peter

When updating or deleting records from sys_sequences table, the built-in RowId column is also affected by changes in sys_sequences table. The changes made to all system table should be made after careful analysis as they can affect library configuration. The built-in rowid column values are matched with user changes in following example.

```
SELECT RowId AS [Built-in RowId], * FROM AutoIncrementTest;
```

	Built-in RowId	Id	Name
▶	1	1	Mary
	3	3	John
	5	5	David
	6	6	Smith
	7	7	Robert
	11	11	Scott
	12	12	Peter

Read the last Identity.

You can read it from SYS_SEQUENCES table to get all the Identities. **Autoincrement Column always returns Int64 or Long.**

```
SELECT * FROM SYS_SEQUENCES;
```

Through .Net Code

```
using (SqlDatabaseConnection sqlcnn = new
SqlDatabaseConnection("schemaName=db;uri=@memory"))
{
    sqlcnn.Open();
    using (SqlDatabaseCommand cmd = new SqlDatabaseCommand(sqlcnn)) {
        cmd.CommandText = "CREATE TABLE IF NOT EXISTS TestTable(Id Integer Primary Key
AutoIncrement, Name Text);";
        cmd.ExecuteNonQuery();

        // Id should be one (1) after first insert
        cmd.CommandText = "INSERT INTO TestTable VALUES(null, 'Hello');";
        cmd.ExecuteNonQuery();

        //LastSequenceNumber requires table name
```

```

Console.WriteLine(string.Format("via cmd.LastSequenceNumber: {0}",
cmd.LastSequenceNumber("TestTable")));

//LastInsertRowId is tracked on connection.
Console.WriteLine(string.Format("via sqlcnnc.LastInsertRowId: {0}",
sqlcnnc.LastInsertRowId));
//last_insert_rowid() is tracked on connection and returns Int64
cmd.CommandText = "SELECT last_insert_rowid()";
Int64 LastID = (Int64)cmd.ExecuteScalar();
Console.WriteLine(string.Format("via SQL: {0}", LastID));

// Id should be two (2) after following insert.
cmd.CommandText = "INSERT INTO TestTable(Name) VALUES('World')";
cmd.ExecuteNonQuery();

//LastSequenceNumber requires table name
Console.WriteLine(string.Format("via cmd.LastSequenceNumber: {0}",
cmd.LastSequenceNumber("TestTable")));

//LastInsertRowId is tracked on connection.
Console.WriteLine(string.Format("via sqlcnnc.LastInsertRowId: {0}",
sqlcnnc.LastInsertRowId));

//last_insert_rowid is tracked on connection SQL Statement
cmd.CommandText = "SELECT last_insert_rowid()";
LastID = (Int64)cmd.ExecuteScalar();
Console.WriteLine(string.Format("via SQL: {0}", LastID));

//Attach another database file to same connection
cmd.CommandText = "ATTACH DATABASE '@memory' AS 'db1'";
cmd.ExecuteNonQuery();
// Create table on schema db1
cmd.CommandText = "CREATE TABLE IF NOT EXISTS db1.TestTable(Id Integer Primary Key
AutoIncrement, Name Text)";
cmd.ExecuteNonQuery();
// Id should be one (1)
cmd.CommandText = "INSERT INTO db1.TestTable VALUES(null, 'Hello')";
cmd.ExecuteNonQuery();
cmd.CommandText = "SELECT last_insert_rowid()";
LastID = (Int64)cmd.ExecuteScalar();
Console.WriteLine(string.Format("via SQL from db1: {0}", LastID));

```

```
cmd.CommandText = "SELECT * FROM db1.TestTable WHERE Id = last_insert_rowid()";
SqlDatabaseDataReader dr = cmd.ExecuteReader();
while (dr.Read())
{
    Console.WriteLine(dr["Name"]);
}
}
```

See also [Sequence_Number Function](#) which can generate numeric sequence.