🛣️

MySQL Commands - CREAT, USE AND DROPING, DELETE, AUTO_INCREMENT, INSERT-IGNORE

DROP DATABASE

```
DROP DATABASE IF EXISTS databasename;
```

DROP TABLE

```
DROP TABLE IF EXISTS schema_name.table_name;
```

DROP SCHEMA

```
DROP SCHEMA IF EXISTS schema_name;
```

SCHEMA IS SYNONYM FOR DATABASE IN MYSQL. SO EITHER USE DATABASE OR SCHEMA BOTH WORK AS SAME.

CREATE DATABASE

```
CREATE DATABASE database_name;
USE database_name;
```

CREATE SCHEMA

```
CREATE SCHEMA schema_name;
USE schema_name;
```

CREATE TABLES

```
-- If written use schema_name first then
CREATE TABLE table_name
(
  -- Skeleton of your table what your table column contain and
  -- what there data type
);

-- If didn't used use scehma first then
CREATE TABLE schema_name.table_name
(
  -- Skeleton of your table what your table column contain and
  -- what there data type
)
```

DELETE FUNCTION

The `DELETE` statement is used to delete existing records in a table.

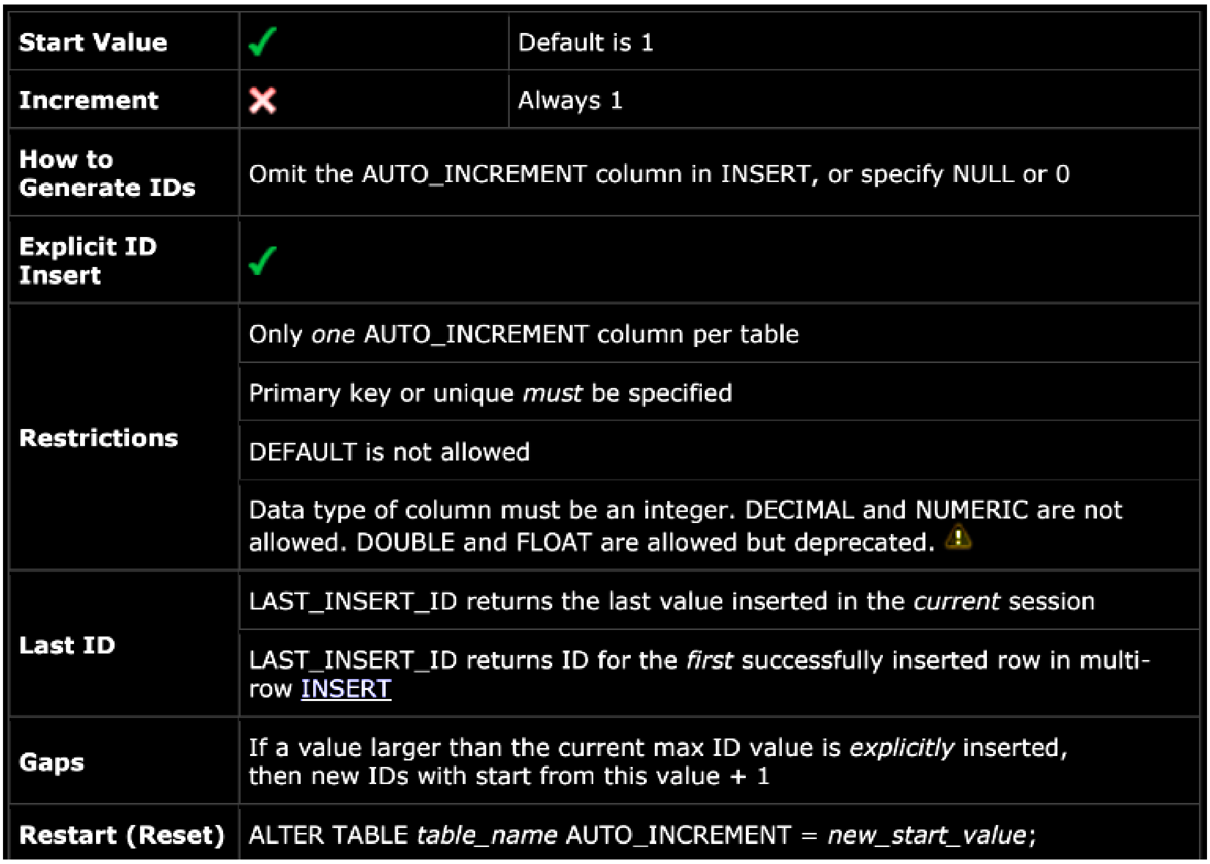
```
DELETE FROM table_name WHERE condition;

-- If you exclude the WHERE clause,
-- all records in the table will be deleted!
```

AUTO INCREMENT

- Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.
- By default, the starting value for `AUTO_INCREMENT` is 1, and it will increment by 1 for each new record.

```
-- Define a table with an auto-increment column (id starts at 100)
CREATE TABLE airlines
(
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(90)
)
AUTO_INCREMENT = 100; -- this change value starting point from 1 to 100
-- Insert a row, ID will be automatically generated
INSERT INTO airlines (name) VALUES ('United Airlines');
-- Get generated ID
SELECT LAST_INSERT_ID();
-- Returns: 100
```



Start Value	✓	Default is 1
Increment	✗	Always 1
How to Generate IDs	Omit the AUTO_INCREMENT column in INSERT, or specify NULL or 0	
Explicit ID Insert	✓	
Restrictions	Only *one* AUTO_INCREMENT column per table	
	Primary key or unique *must* be specified	
	DEFAULT is not allowed	
	Data type of column must be an integer. DECIMAL and NUMERIC are not allowed. DOUBLE and FLOAT are allowed but deprecated. ⚠	
Last ID	LAST_INSERT_ID returns the last value inserted in the *current* session	
	LAST_INSERT_ID returns ID for the *first* successfully inserted row in multi-row INSERT	
Gaps	If a value larger than the current max ID value is *explicitly* inserted, then new IDs with start from this value + 1	
Restart (Reset)	ALTER TABLE *table_name* AUTO_INCREMENT = *new_start_value*;	

- To let the `AUTO_INCREMENT` sequence start with another value, use the following SQL statement:

```
ALTER TABLE airlines AUTO_INCREMENT=2;
```

- To generate a ID value, you can exclude the auto-increment column in INSERT statement, or specify `NULL` or `0` value explicitly:

  ```
  -- Omit auto-increment column
  INSERT INTO airlines (name) VALUES ('Delta');
  -- Specify NULL or 0
  INSERT INTO airlines VALUES (NULL, 'Southwest');
  INSERT INTO airlines VALUES (0, 'American Airlines');
  ```

MAKE A GAP

You can insert an ID value explicitly, then MySQL will generate new IDs starting from it adding 1:

```
INSERT INTO airlines VALUES (200, 'Lufthansa');
INSERT INTO airlines (name) VALUES ('British Airways');
-- id 201(LAST_ID IN TABLE)  is assigned
```

You can still insert inside the gap using ID less than the current maximum ID, but this does not affect ID that will be used for other rows:

```
INSERT INTO airlines VALUES (150, 'Air France'); -- id 150 inserted
INSERT INTO airlines (name) VALUES ('KLM');
-- id 202(LAST_ID IN TABLE)  is assigned
```

RESTART ID

You cannot reset the auto-increment counter to the start value less or equal than the current maximum ID:

```
ALTER TABLE airlines AUTO_INCREMENT = 1;
INSERT INTO airlines (name) VALUES ('US Airways');
-- id 205(LAST_ID IN TABLE) is assigned
```

After you have deleted all rows, the counter is not automatically reset to the start value:

```
DELETE FROM airlines;
INSERT INTO airlines (name) VALUES ('United'); -- id 206 is assigned
```

You can restart the auto-increment to 1 if there are no rows in a table:

```
DELETE FROM airlines;
ALTER TABLE airlines AUTO_INCREMENT = 1;
INSERT INTO airlines (name) VALUES ('United'); -- id 1 is assigned
```

You can use auto increment when there is no data in the table. If you use it on filled table it will throw you an error.

INSERT IGNORE

When you use the `INSERT` statement to add multiple rows to a table and if an error occurs during the processing, MySQL terminates the statement and returns an error. As the result, no rows are inserted into the table.

However, if you use the `INSERT IGNORE` statement, the rows with invalid data that cause the error are ignored and the rows with valid data are inserted into the table.

```
CREATE TABLE subscribers (
    id INT PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(50) NOT NULL UNIQUE
);
```

```
INSERT INTO subscribers(email)
VALUES('john.doe@gmail.com');
```

```
-- THIS WILL EXECUTE PROPERLY.
```

```
INSERT INTO subscribers(email)
VALUES('john.doe@gmail.com'),
      ('jane.smith@ibm.com');

-- It returns an error.
-- Error: Duplicate entry 'john.doe@gmail.com' for key 'email'
```

However, if you use the `INSERT IGNORE` statement instead.

```
INSERT IGNORE INTO subscribers(email)
VALUES('john.doe@gmail.com'),
      ('jane.smith@ibm.com');

-- 1 row(s) affected, 1 warning(s): 1062 Duplicate entry
-- 'john.doe@gmail.com' for key 'email' Records: 2
-- Duplicates: 1  Warnings: 1
```