Table of Contents

1. Insert . . . on Duplicate Key Update

MySQL supports a version of the insert statement that combines the options of insert and update.

Assume that col_id was set as the primary key of the table. This is the statement model:

```
Insert into TblName ( col_id, col_2, col_3,. . . )
   values (val_id, val_2, val_3,  . . . )
   on duplicate key update
   set col_2 = val_2, col_3 = val_3, . . .
```

The `on duplicate key update` clause is available for `Insert Into tblName Values`, `Insert Into tblName Set` and the `Insert Into tblName Select` variation of the Insert.

Demo 01: Set up a new table containing the following rows: (use the sql in the demo.)

```
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Frosting    |       NULL |
|     30 | Sphinx      |       NULL |
|     40 | Kilimanjaro |       NULL |
+--------+-------------+------------+
```

Demo 02: If we try to do another insert using the value 20 for the project id, we get a duplicate key error.

```
insert into ac_projects
values  (20, 'Icing', 450000);
ERROR 1062 (23000): Duplicate entry '20' for key 'PRIMARY'
```

Demo 03: We can use the following which says what to do if that would make a duplicate key. In that case we want to update the other columns.

```
insert into ac_projects
values  (20, 'Icing', 450000)
on duplicate key update
  prj_name  = 'Icing'
, prj_budget = 450000;
Query OK, 2 rows affected (0.05 sec)
```

MySQL reports back that two rows were affected. And the data becomes as shown here. Note that the previous row for prj_id has been updated to the newly supplied values.

```
select * from ac_projects;
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Icing       |     450000 |
|     30 | Sphinx      |       NULL |
|     40 | Kilimanjaro |       NULL |
+--------+-------------+------------+
```

Demo 04: This one does an insert since it is a new id

```
insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('cupcake', 42, 12000)
on duplicate key update
  prj_name = 'cupcake'
, prj_budget = 50000;
select * from ac_projects;
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Icing       |     450000 |
|     30 | Sphinx      |       NULL |
|     40 | Kilimanjaro |       NULL |
|     42 | cupcake     |      12000 |
+--------+-------------+------------+
```

This can be a useful command variation or a dangerous one. If you think of an insert statement as adding a new row, then you need to be aware that this may insert or may update. That is a different way of thinking about a command.

1.1. Some variations

Demo 05: Using values() to refer to the column value from the insert

```
insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('Catapult', 50, 586000)
on duplicate key update
  prj_name = values(prj_name)
, prj_budget = values(prj_budget);
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Icing       |     450000 |
|     30 | Sphinx      |       NULL |
|     40 | Kilimanjaro |       NULL |
|     42 | cupcake     |      12000 |
|     50 | Catapult    |     586000 |
+--------+-------------+------------+
6 rows in set (0.00 sec)

insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('Crescent', 30, 4000)
on duplicate key update
  prj_name = values(prj_name)
, prj_budget = values(prj_budget);
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Icing       |     450000 |
|     30 | Crescent    |       4000 |
|     40 | Kilimanjaro |       NULL |
|     42 | cupcake     |      12000 |
|     50 | Catapult    |     586000 |
+--------+-------------+------------+
```

Demo 06: The update expression can use the current value of the row as shown below. The value of the budget will be updated to the larger of the current row or the proposed value. (Greatest is a single row function; it is not the same as the Max aggregate function)

```
insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('Crescent', 30, 2500)
on duplicate key update
  prj_name = values(prj_name)
, prj_budget =greatest( ac_projects.prj_budget, values(prj_budget))
;
select * from ac_projects;
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Icing       |     450000 |
|     30 | Crescent    |       4000 |
|     40 | Kilimanjaro |       NULL |
|     42 | cupcake     |      12000 |
|     50 | Catapult    |     586000 |
+--------+-------------+------------+
insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('Crescent', 30, 7500)
on duplicate key update
  prj_name = values(prj_name)
, prj_budget =greatest( ac_projects.prj_budget, values(prj_budget))
;
select * from ac_projects;
+--------+-------------+------------+
| prj_id | prj_name    | prj_budget |
+--------+-------------+------------+
|     10 | Snowball    |       NULL |
|     20 | Icing       |     450000 |
|     30 | Crescent    |       7500 |
|     40 | Kilimanjaro |       NULL |
|     42 | cupcake     |      12000 |
|     50 | Catapult    |     586000 |
+--------+-------------+------------+
```

Demo 07: What about a multi-row insert?

```
insert into ac_projects (prj_name, prj_id, prj_budget) values
  ('Godot',     101,   50001)
, ('Milan',     102,   50002)
, ('Omega',     030,  125000)
, ('Palladium', 050,   50003)
, ('Greenwich', 103,   50004)
, ('Deco',      040,   50005)
, ('Volta',     042,   50005)
on duplicate key update
  prj_name   = values(prj_name)
, prj_budget = greatest( ac_projects.prj_budget, values(prj_budget)
);
+--------+-----------+------------+
| prj_id | prj_name  | prj_budget |
+--------+-----------+------------+
|     10 | Snowball  |       NULL |
```

```
|     20 | Icing     |     450000 |
|     30 | Omega     |     125000 |
|     40 | Deco      |       NULL |
|     42 | Volta     |      50005 |
|     50 | Palladium |     586000 |
|    101 | Godot     |      50001 |
|    102 | Milan     |      50002 |
|    103 | Greenwich |      50004 |
+--------+-----------+------------+
```

Demo 08: New row or increase the budget if the dept has a budget value.

```
insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('Volta', 042, null)
on duplicate key update
 prj_budget = ac_projects.prj_budget * 1.5
;


insert into ac_projects (prj_name, prj_id, prj_budget)
values  ('Volta', 043, null)
on duplicate key update
 prj_budget = ac_projects.prj_budget * 1.5
;
+--------+-----------+------------+
| prj_id | prj_name  | prj_budget |
+--------+-----------+------------+
|     10 | Snowball  |       NULL |
|     20 | Icing     |     450000 |
|     30 | Omega     |     125000 |
|     40 | Deco      |       NULL |
|     42 | Volta     |      75008 |
|     43 | Volta     |       NULL |
|     50 | Palladium |     586000 |
|    101 | Godot     |      50001 |
|    102 | Milan     |      50002 |
|    103 | Greenwich |      50004 |
+--------+-----------+------------+
```