Difference between On Delete Cascade & On Update Cascade in mysql

Asked 6 years, 9 months ago Active 3 years, 6 months ago Viewed 179k times

79

31

I have two tables in MySQL database- parent , child . I'm trying to add foreign key references to my child table based on the parent table. Is there any significant difference between ON UPDATE CASCADE and ON DELETE CASCADE

My Parent Table

```
CREATE TABLE parent (
  id INT NOT NULL,
  PRIMARY KEY (id)
) ENGINE=INNODB;
```

My Question is: What is the difference between the following sql queries.

1. **ON DELETE CASCADE**

```
CREATE TABLE child (
  id INT,
  parent_id INT,
  INDEX par_ind (parent_id),
  FOREIGN KEY (parent_id)
    REFERENCES parent(id)
    ON DELETE CASCADE
) ENGINE=INNODB;
```

2. **ON UPDATE CASCADE**

```
CREATE TABLE child (
  id INT,
  parent_id INT,
  INDEX par_ind (parent_id),
  FOREIGN KEY (parent_id)
    REFERENCES parent(id)
    ON UPDATE CASCADE
) ENGINE=INNODB;
```

3. **ON UPDATE CASCADE ON DELETE CASCADE**

```
CREATE TABLE child (
    id INT,
    parent_id INT,
    INDEX par_ind (parent_id),
    FOREIGN KEY (parent_id)
      REFERENCES parent(id)
      ON UPDATE CASCADE ON DELETE CASCADE
) ENGINE=INNODB;
```

Are there any errors in the queries? What do these queries (1,2 & 3) mean?? Are they same???

mysql innodb mysql-5.5 foreign-key

Share Improve this question Follow

edited Nov 22 '17 at 18:55 Israel is an Apartheid State 50.3k 32 178 375

asked Aug 22 '14 at 6:22 Lonewolf 901 1 7 8

2 p.s. <nitpick> for completeness, what you are talking about above are DDL (Data Definition Language) **statements**, and not queries. A query is generally considered to be DML (Data Manipulation Language SELECT, INSERT, UPDATE, DELETE) </nitpick> – Vérace Aug 22 '14 at 9:34

Another p.s. again for completeness, I wondered what the default was. So I created a child with no on update or on delete. What happens then is that you can neither update nor delete a parent that has a dependent child. That makes perfect sense, however MySQL is not always a model of that particular characteristic :-) – Vérace Aug 22 '14 at 12:53

Add a comment

3 Answers

Active Oldest Votes

110

A very good thread on this subject is to be found here and also here. The definitive guide for MySQL is, of course, the documentation, to be found here.

In the SQL 2003 standard there are 5 different referential actions:

1. CASCADE
2. RESTRICT
3. NO ACTION
4. SET NULL
5. SET DEFAULT

To answer the question:

1. **CASCADE**
   - ON DELETE CASCADE means that if the parent record is deleted, any child records are also deleted. This is **not** a good idea in my opinion. You should keep track of all data that's ever been in a database, although this can be done using TRIGGER s. (However, see caveat in comments below).
   - ON UPDATE CASCADE means that if the parent primary key is changed, the child value will also change to reflect that. Again in my opinion, not a great idea. If you're changing PRIMARY KEY s with any regularity (or even at all!), there is something wrong with your design. Again, see comments.
   - ON UPDATE CASCADE ON DELETE CASCADE means that if you UPDATE **OR** DELETE the parent, the change is cascaded to the child. This is the equivalent of AND ing the outcomes of first two statements.
2. **RESTRICT**
   - RESTRICT means that any attempt to delete and/or update the parent will fail throwing an error. This is the default behaviour in the event that a referential action is not explicitly specified.

     For an ON DELETE or ON UPDATE that is not specified, the default action is always RESTRICT'.
3. **NO ACTION**
   - NO ACTION : From the manual. A keyword from standard SQL. In MySQL, equivalent to RESTRICT . The MySQL Server rejects the delete or update operation for the parent table if there is a related foreign key value in the referenced table. Some database systems have deferred checks, and NO ACTION is a deferred check. In MySQL, foreign key constraints are checked immediately, so NO ACTION is the same as RESTRICT .
4. **SET NULL**
   - SET NULL - again from the manual. Delete or update the row from the parent table, and set the foreign key column or columns in the child table to NULL . This is not the best of ideas IMHO, primarily because there is no way of "time-travelling" - i.e. looking back into the child tables and associating records with NULL s with the relevant parent record - either CASCADE or use TRIGGER s to populate logging tables to track changes (but, see comments).
5. **SET DEFAULT**
   - SET DEFAULT . Yet another (potentially very useful) part of the SQL standard that MySQL hasn't bothered

implementing! Allows the developer to specify a value to which to set the foreign key column(s) on an UPDATE or a DELETE. InnoDB and NDB will reject table definitions with a `SET DEFAULT` clause.

As mentioned above, you should spend some time looking at the documentation, here.

Share Improve this answer Follow

edited Nov 22 '17 at 18:53
Israel is an Apartheid State
50.3k 32 178 375

answered Aug 22 '14 at 7:49
Vérace
23.7k 5 53 70

14 I like your complete answer however I disagree with this statement. "You should keep track of all data that's ever been in a database" - this is really dependent upon the design and purposes of the database. For example a Recipe Definition (I am not talking food - more like systems configurations) when the recipe definition is deleted it makes no sense to keep the associated children of that recipe - that just bloats the db for no reason. Also work tables for machine systems - I do not need the data anymore; process and get rid of it. Other than that your answer is fantastic. – StixO Aug 15 '16 at 13:48

3 similar to @StixO I like this answer mostly, but I gotta disagree with changing primary key. There are definitely designs where this would be a bad idea but when you get into a distributed database it can be very desirable that primary keys are free to be reassigned without losing the identity of a record. – Garet Claborn Aug 11 '17 at 22:50

1 "This is not a good idea in my opinion. You should keep track of all data that's ever been in a database." - Not sure I understand your point. If you are cascading 'on delete', then you have already decided you need to delete something. If you do decide to never delete anything, nothing will cascade. The benefit of having it though is that in your application you can be sure that when you look for a record with a foreign ID, you know that it will be there, and there wont be any orphan rows bloating your database should you decide to delete something. – Jeff Ryan Dec 10 '18 at 0:42

1 The logic here is pretty faulty in places, and even more so in our new GDPR world. I do agree with the notion that if primary keys are changing, it might be a sign of something wrong though. – Chuck Le Butt Feb 19 '19 at 11:44

If you're changing PRIMARY KEYs with any regularity (or even at all!), there is something wrong with your design. Do you mean ON UPDATE CASCADE changes the value of the key or the name of the key? – Billal Begueradj Mar 4 '19 at 11:09

Add a comment

9

These two are actions to be performed, respectively, when the referenced record on the parent table changes its id and when it gets deleted.

If you execute:

```
UPDATE parent SET id = -1 WHERE id = 1;
```

And there is at least one record on `child` with `parent_id = 1` , 1) will fail; in cases 2) and 3), all records with parent_id = 1 are updated to parent_id = -1.

If you execute:

```
DELETE FROM parent WHERE id = 1;
```

And there is at least one record on `child` with `parent_id = 1` , 2) will fail; in cases 1) and 3), all records with `parent_id = 1` are deleted.

3) is syntactically correct.

Full documentation can be found on the manual.

Share Improve this answer Follow

answered Aug 22 '14 at 8:02
jynus
13.6k 1 26 37

2 Answer would be better served by defining 1,2,3 rather than relying on the question never being edited. – iheanyi Mar 4 '20 at 0:27

Add a comment

7

I don't have enough reputation to comment on the previous answers. So I thought I'd elaborate a bit.

1) ON DELETE CASCADE means if the parent record is deleted, then any referencing child records are also deleted. ON UPDATE defaults to RESTRICT, which means the UPDATE on the parent record will fail.

2) ON DELETE action defaults to RESTRICT, which means the DELETE on the parent record will fail. ON UPDATE CASCADE will update all referencing child records when the parent record is updated.

3) See the CASCADE actions in 1) and 2) above.

On using parent record IDs as foreign keys (in child tables) -- experience says a) if the IDs are auto-generated sequence numbers, then DO NOT use them as foreign keys. Use some other unique parent key instead. b) if the IDs are GUIDs, then it's ok to use them as foreign keys. You'll see the wisdom in this suggestion when you export and import the records or copy records to another database. It's too cumbersome to deal with auto-generated sequence numbers during data migration when they are referenced as foreign keys.

Share Improve this answer Follow

answered May 5 '15 at 16:34
g r
71 1 1

Add a comment

Your Answer

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post as a guest

Name

Email

Required, but never shown

By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy

Not the answer you're looking for? Browse other questions tagged mysql innodb mysql-5.5 foreign-key or ask your own question.