

# On\_delete Behaviors

---

[source: stackoverflow](#)

指当所引用的对象被删除时，对 该对象的引用者的处理方式

It is not specific to django, this is an **SQL standard**.

有6种处理方式:

- **CASCADE**:
  - 同时删除引用者
  - (When you remove a blog post for instance, you might want to delete comments as well).
  - SQL equivalent: **CASCADE**.
- **PROTECT**:
  - 禁止对所引用对象的删除行为。
  - 若想删除一个引用对象，必须先删除其所有引用者。
  - SQL equivalent: **RESTRICT**.
- **SET\_NULL**:
  - 将引用对象设NULL (requires the field to be nullable).
  - For instance, when you delete a User, you might want to keep the comments he posted on blog posts, but say it was posted by an anonymous (or deleted) user.
  - SQL equivalent: **SET NULL**.
- **SET\_DEFAULT**:
  - 将引用对象设为default值。
  - SQL equivalent: **SET DEFAULT**.
- **SET(...)**:
  - 将引用对象设为特定值
  - 这是Django的特有行为，并不是SQL标准。
- **DO\_NOTHING**:
  - 什么都不做
  - Probably a very bad idea since this would create integrity issues in your database (referencing an object that actually doesn't exist).
  - SQL equivalent: **NO ACTION**.

**Note:** `on_delete` will become a required argument in Django 2.0. In older versions it defaults to **CASCADE**.