


# Atomic Transactions

```
with transaction.atomic():  
    thing = SomeModel.objects.create(...)  
    other_thing = SomeModel.objects.create(...)  
  
    if error_condition(...):  
        raise Exception('Rolls back entire transaction')
```



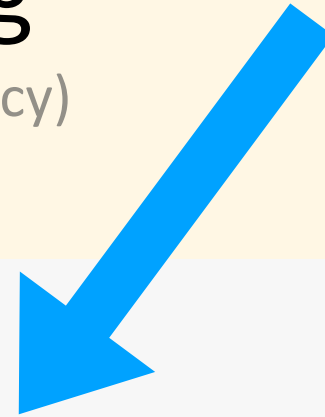
Exceptions roll back the entire transaction block.

Neither object will be saved to the DB.

Transactions can be nested.

# Row Locking

(pessimistic concurrency)



```
with transaction.atomic():  
    to_update = SomeModel.objects.select_for_update().filter(id=thing.id)  
    ...  
    to_update.update(val=new)
```

`.select_for_update()` allows you to lock rows

Locking prevents other threads from changing the row until the end of the current transaction, when the lock is released.