```php
<?php
//@source https://coderwall.com/p/rml5fa/nested-pdo-transactions

/**
 * This class extends native PDO one but allow nested transactions
 * by using the SQL statements `SAVEPOINT', 'RELEASE SAVEPOINT' AND 'ROLLBACK
SAVEPOINT'
 */
class ExtendedPdo extends PDO
{
  /**
   * @var array Database drivers that support SAVEPOINT * statements.
   */
  protected static $_supportedDrivers = array("pgsql", "mysql");

  /**
   * @var int the current transaction depth
   */
  protected $_transactionDepth = 0;


  /**
   * Test if database driver support savepoints
   *
   * @return bool
   */
  protected function hasSavepoint()
  {
    return in_array($this->getAttribute(PDO::ATTR_DRIVER_NAME),
      self::$_supportedDrivers);
  }


  /**
   * Start transaction
   *
   * @return bool|void
   */
  public function beginTransaction()
  {
    if($this->_transactionDepth == 0 || !$this->hasSavepoint()) {
      parent::beginTransaction();
    } else {
      $this->exec("SAVEPOINT LEVEL{$this->_transactionDepth}");
    }

    $this->_transactionDepth++;
  }

  /**
   * Commit current transaction
   *
   * @return bool|void
   */
```

```php
54   public function commit()
55   {
56     $this->_transactionDepth--;
57
58     if($this->_transactionDepth == 0 || !$this->hasSavepoint()) {
59       parent::commit();
60     } else {
61       $this->exec("RELEASE SAVEPOINT LEVEL{$this->_transactionDepth}");
62     }
63   }
64
65   /**
66    * Rollback current transaction,
67    *
68    * @throws PDOException if there is no transaction started
69    * @return bool|void
70    */
71   public function rollBack()
72   {
73
74     if ($this->_transactionDepth == 0) {
75       throw new PDOException('Rollback error : There is no transaction started');
76     }
77
78     $this->_transactionDepth--;
79
80     if($this->_transactionDepth == 0 || !$this->hasSavepoint()) {
81       parent::rollBack();
82     } else {
83       $this->exec("ROLLBACK TO SAVEPOINT LEVEL{$this->_transactionDepth}");
84     }
85   }
86
87 }
```