MySQL 事务：

（1）在 MySQL 中只有使用了 **Innodb 数据库引擎**的数据库或表才**支持事务**；

（2）事务处理可以用来**维护数据库的完整性**，保证成批的 SQL 语句要么全部执行，要么全部不执行；

（3）事务用来管理 insert,update,delete 语句。

一般来说，事务是必须满足 4 个条件（ACID）：原子性（Atomicity，或称不可分割性）、一致性（Consistency）、隔离性（Isolation，又称独立性）、持久性（Durability）。

（1）原子性：一个事务（transaction）中的所有操作，**要么全部完成，要么全部不完成**，不会结束在中间某个环节。事务在执行过程中发生错误，会被回滚（Rollback）到事务开始前的状态，就像这个事务从来没有执行过一样；

（2）一致性：在事务开始之前和事务结束以后，数据库的完整性没有被破坏。这表示写入的资料必须完全符合所有的预设规则，这包含资料的精确度、串联性以及后续数据库可以自发性地完成预定的工作；

（3）隔离性：数据库允许多个并发事务同时对其数据进行读写和修改的能力，隔离性可以防止多个事务并发执行时由于交叉执行而导致数据的不一致。事务隔离分为不同级别，包括读未提交（Read uncommitted）、读提交（read committed）、可重复读（repeatable read）和串行化（Serializable）；

（4）持久性：事务处理结束后，对数据的修改就是永久的，即便系统故障也不会丢失。

事务控制语句：

（1）BEGIN 或 START TRANSACTION 显式地**开启一个事务**；

（2）COMMIT 也可以使用 COMMIT WORK，不过二者是等价的。COMMIT 会**提交事务**，并使已对数据库进行的所有修改成为**永久性**的；

（3）ROLLBACK 也可以使用 ROLLBACK WORK，不过二者是等价的。会滚回结束用户的事务，并撤销正在进行的所有未提交的修改；

（4）SAVEPOINT identifier，SAVEPOINT 允许在事务中创建一个保存点，一个事务中可以有多个 SAVEPOINT；

（5）RELEASE SAVEPOINT identifier 删除一个事务的保存点，当没有指定的保存点时，执行该语句会抛出一个异常；

（6）ROLLBACK TO identifier 把事务回滚到标记点；

（7）SET TRANSACTION 用来设置事务的隔离级别。InnoDB 存储引擎提供事务的隔离级别有 READ UNCOMMITTED、READ COMMITTED、REPEATABLE READ 和 SERIALIZABLE。

MySQL 事务处理主要有两种方法：

1. 用 BEGIN, ROLLBACK, COMMIT 来实现

（1）BEGIN 开始一个事务；

（2）ROLLBACK 事务回滚；

（3）COMMIT 事务确认。

2. 直接用 SET 来改变 MySQL 的自动提交模式

（1）SET AUTOCOMMIT=0 禁止自动提交；

（2）SET AUTOCOMMIT=1 开启自动提交。

实例：

```
mysql> use RUNOOB;
Database changed
mysql> CREATE TABLE runoob_transaction_test( id int(5)) engine=innodb;  # 创建数据表
Query OK, 0 rows affected (0.04 sec)

mysql> select * from runoob_transaction_test;
Empty set (0.01 sec)

mysql> begin;  # 开始事务
Query OK, 0 rows affected (0.00 sec)

mysql> insert into runoob_transaction_test value(5);
Query OK, 1 rows affected (0.01 sec)

mysql> insert into runoob_transaction_test value(6);
Query OK, 1 rows affected (0.00 sec)

mysql> commit; #  提交事务
Query OK, 0 rows affected (0.01 sec)

mysql>    select * from runoob_transaction_test;
+------+
| id   |
+------+
| 5    |
| 6    |
+------+
2 rows in set (0.01 sec)

mysql> begin;      # 开始事务
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>    insert into runoob_transaction_test values(7);
Query OK, 1 rows affected (0.00 sec)

mysql> rollback;     # 回滚
Query OK, 0 rows affected (0.00 sec)

mysql>     select * from runoob_transaction_test;     # 因为回滚所以数据没有插入
+------+
| id   |
+------+
| 5    |
| 6    |
+------+
2 rows in set (0.01 sec)
```

MySQL ALTER 命令：

当需要修改数据表名或者修改数据表字段时，就需要用到 MySQL ALTER 命令。

删除、添加或修改表字段：

如下命令使用了 ALTER 命令及 DROP 子句来删除以上创建表的 i 字段：

```
mysql> ALTER TABLE testalter_tbl   DROP i;
```

如果需要指定新增字段的位置，可以使用 MySQL 提供的关键字 FIRST (设定位第一列)， AFTER 字段名（设定位于某个字段之后）。

```
ALTER TABLE testalter_tbl DROP i;
ALTER TABLE testalter_tbl ADD i INT FIRST;
ALTER TABLE testalter_tbl DROP i;
ALTER TABLE testalter_tbl ADD i INT AFTER c;
```

FIRST 和 AFTER 关键字可用于 ADD 与 MODIFY 子句，所以如果你想重置数据表字段的位置就需要先使用 DROP 删除字段然后使用 ADD 来添加字段并设置位置。

修改字段类型及名称：

如果需要修改字段类型及名称，你可以在 ALTER 命令中使用 MODIFY 或 CHANGE 子句。

例如，把字段 c 的类型从 CHAR(1) 改为 CHAR(10)，可以执行以下命令:

```
mysql> ALTER TABLE testalter_tbl MODIFY c CHAR(10);
```

使用 CHANGE 子句, 语法有很大的不同。在 CHANGE 关键字之后，紧跟着的是你要修改的字段名，然后指定新字段名及类型。尝试如下实例：

mysql> ALTER TABLE testalter_tbl CHANGE i j BIGINT;
mysql> ALTER TABLE testalter_tbl CHANGE j j INT;

ALTER TABLE 对 NULL 值和默认值的影响：

当修改字段时，可以指定是否包含值或者是否设置默认值。

以下实例，指定字段 j 为 NOT NULL 且默认值为 100。

```
mysql> ALTER TABLE testalter_tbl
    -> MODIFY j BIGINT NOT NULL DEFAULT 100;
```

如果不设置默认值，MySQL 会自动设置该字段默认为 NULL。

修改字段默认值：

mysql> ALTER TABLE testalter_tbl ALTER i SET DEFAULT 1000;

也可以使用 ALTER 命令及 DROP 子句来删除字段的默认值，如下实例：

mysql> ALTER TABLE testalter_tbl ALTER i DROP DEFAULT;

修改数据表类型，可以使用 ALTER 命令及 TYPE 子句来完成。尝试以下实例，我们将表 testalter_tbl 的类型修改为 MYISAM：

mysql> ALTER TABLE testalter_tbl ENGINE = MYISAM;
mysql> SHOW TABLE STATUS LIKE 'testalter_tbl'\G

修改表名：

在 ALTER TABLE 语句中使用 RENAME 子句来实现。

mysql> ALTER TABLE testalter_tbl RENAME TO alter_tbl;