

数据库的安全性是指保证数据不被非法访问，保证数据不会因非法使用而被泄密、更改或破坏。

事务是一系列数据库操作的有限序列，是数据库的基本执行单元。事务的根本特征是，其包含的操作序列要么全做，要么全不做，整个序列是一个不可分割的整体。

每个事务可以看成数据库的一个状态，而整个应用程序的操作过程则是通过不同事务使得数据库从一个状态变换成另一个状态。

事务的性质：

原子性：事务必须是数据库的逻辑工作单元，即食物中包括的诸操作要么全执行，要么全不执行。

一致性：事务在完成时，必须使所有数据都保持一致状态。一致状态：只含成功事务提交的结果。不一致状态：包含失败事务的结果

隔离性：一个事务的执行不能被其他事务干扰，即一个事务内部的操作及使用的数据对其他的并发事务是隔离的，并发执行的各个事务间不能互相干扰。

持久性：指一个事务一旦提交，则它对数据库中的数据改变就应该是永久的。

事务并发执行时的数据访问冲突：丢失更新、读脏数据、不可重复读。

事务并发执行时的数据访问冲突：丢失更新、读脏数据、不可重复读。

丢失更新：当两个事务或多个事务选择同一行时，然后计算最初选定的值更新该行时，由于每个事务都不知道其他事务的存在，因此最后的更新将重写有其他事务所进行的更新，这将导致前面的事务更新的数据丢失。

读脏数据：事务 T1 修改数据，并将其写回；事务 T2 读取了该数据，但 T1 随后又因某种原因撤销了，使得 T2 读取的数据与数据库中的数据不一致。即 T2 读取的是“脏”（不正确）数据

不可重复读：当事务 T1 读取某数据后，事务 T2 对该数据执行了更新操作，使得 T1 无法再次读取与上一次相同的数据。这种数据不一致称为不可重复读。

对于一个并发事务集，如果一个调度与一个串行调度等价，则称该调度是可串行化的。按可串行化调度的事务的执行，称为并发事务的可串行化。使并发调度可串行化的技术称为并发控制技术。在一般的 DBMS 中，都以可串行化作为并发控制的正确性准则。

事物的并发控制要对多事务并发执行中的所有操作按正确方式进行调度，使得一个事务的执行不受其他事务的干扰。并发控制的主要技术是封锁机制。

常见锁有排他锁和共享锁。

排他锁（写锁或 X 锁）：若事务 T 对数据 A 加上排他锁，则 T 可对 A 进行读写，其他事务只有等到 T 解除对 A 的封锁后，才能对 A 进行封锁和操作。

共享锁（读锁或 S 锁）：若事务 T 对数据 A 加上共享锁，则 T 对 A 只能读取不能修改，其他事务可对 A 加 S 锁，但不能加 X 锁

封锁协议：

- 一级封锁协议：事务 **T** 在对数据 **A** 进行写操作前，必须对 **A** 加 **X** 锁，直到事务结束，可防止丢失更新。
- 二级封锁协议：事务 **T** 在读数据 **A** 之前必须先对 **A** 加 **S** 锁，读完后立即释放 **S** 锁，可以防止丢失更新和读脏数据。
- 三级封锁协议：事务 **T** 在读数据 **A** 之前必须先对 **A** 加 **S** 锁，事务完成后才释放 **S** 锁，可以防止丢失更新、读脏数据和不可重复读问题。