Transactions

Transaction Mask

Transaction Number Column

Active Transactions

An active transaction is a transaction that is currently being executed in the database. Checking the transactions in a database is **very helpful to verify to maintain consistency** and the integrity of a database. While a transaction is active, the database locks the resources it needs to complete the transaction to prevent other transactions from modifying the same data. **You should manage transactions properly, as long-running transactions can lead to performance problems.**

If a transaction is active, **one of the following conditions must be true:**

- The transaction has not yet started.
- The transaction has started but is not yet completed.
- The transaction has started, but could not however complete successfully, due to, for example, a system crash or communication problems, etc…

The status of each transaction is recorded in the Transaction Inventory Page (TIP). In fact, **committing a transaction just changes the status of the transaction from active to committed in the TIP.**

Transactions in limbo

Firebird/InterBase®'s transaction mechanism, like most databases, can only handle transactions within a single database. However, within an embedded SQL application, Firebird/InterBase® can perform operations on more than one database at a time.

Transactions in limbo are transactions that have been initiated but have not yet been committed or rolled back. This can happen when a transaction involves multiple nodes or resources within the distributed database system, and a failure or network issue occurs before the transaction can be completed. When this happens, the transaction is left in a state of uncertainty or limbo, as the system is unsure whether the transaction has been successfully committed or not.

Transactions in limbo are transactions that have been started by the PREPARE command within the framework of a two-phase commit. The two-phase commit protocol resolves transactions in limbo, which makes sure that all nodes involved in the transaction can agree on whether to commit or roll back the transaction. The first part of the protocol entails preparing all nodes for the transaction, and the second phase involves either committing or rolling back the transaction based on the outcome of the first phase.

For a database backup to be fully performed while aborting, such transactions in limbo need to be ignored in the backup. Only the most recent, committed transactions are backed up. It allows a database to be backed up, before recovering corrupted transactions. Generally in limbo transactions should be recovered before a backup is performed.